

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF RADIO ELECTRONICS



TECHNOLOGIE IEEE802.15.4, ZIGBEE A PŘÍKLADY JEJICH APLIKACÍ

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. VOJTĚCH SKLÁDANÝ

BRNO 2008

	<p>VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ BRNO UNIVERSITY OF TECHNOLOGY</p>
	<p>FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ ÚSTAV RADIOELEKTRONIKY</p> <p>FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION DEPARTMENT OF RADIO ELECTRONICS</p>

TECHNOLOGIE IEEE802.15.4, ZIGBEE A PŘÍKLADY JEJICH APLIKACÍ

IEEE802.15.4 AND ZIGBEE TECHNOLOGIES, EXAMPLES OF APPLICATIONS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

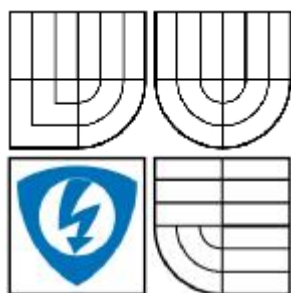
AUTOR PRÁCE
AUTHOR

Bc. VOJTĚCH SKLÁDANÝ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MICHAL KOVÁČ

BRNO 2008



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav radioelektroniky

Diplomová práce

magisterský navazující studijní obor

Elektronika a sdělovací technika

Student: Skládaný Vojtěch Bc.

ID: 89795

Ročník: 2

Akademický rok: 2007/2008

NÁZEV TÉMATU:

Technologie IEEE802.15.4, ZigBee a příklady jejích aplikací

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte a osvojte si základy komunikačních technologií IEEE802.15.4 a ZigBee.

Navrhnete aplikace pro zjednodušenou IEEE802.15.4 MAC vrstvu (SMAC Freescale) na testovacích modulech s procesory HCS08. Rozšířte SMAC na procesory ColdFire a ověřte vzájemnou komunikaci.

Prostudujte a osvojte si kompletní vrstvu IEEE802.15.4. Navrhnete aplikace pro procesory Freescale. Ověřte kompatibilitu jiných implementací (TI, Atmel, OKI).

Jako aplikaci realizujte nástroj pro projektování a instalaci radiových sítí podle standardu IEEE802.15.4. Koncepce bude sestavena z vysílače s bateriovým napájením, který bude trvale posílat zprávy, a přijímače vybaveným řádkovým LCD displejem, kterého MCU vyhodnocuje zprávy z modulu Maxstream a Meshnetics. Displej bude zobrazovat parametry přijímaného signálu jako jsou síla signálu, četnost chyb atd.

DOPORUČENÁ LITERATURA:

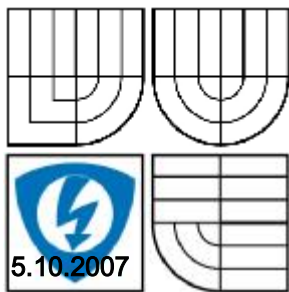
[1] PATRICK, D. Wireless Network Coexistence. Englewood Cliffs: McGraw-Hill Professional Publishing, 2004. ISBN 0071399151

[2] GUTIERREZ, CALLAWAY, BARRETT. Enabling Wireless Sensors with IEEE 802.15.4, Low-Rate Wireless Personal Area Networks. Piscataway: IEEE Press, 2003. ISBN 0738135577

[3] Aplikační zprávy firmy Freescale (www.freescale.com)

UPOZORNĚNÍ:

Další literatura a dokumentace bude dostupná ve firmě Beta Control Brno. Technický vedoucí práce je autor. Studenti a uchazeči nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav radioelektroniky

Termín zadání:

5.10.2007

Termín odevzdání: 30.5.2008

Vedoucí práce:

Ing. Michal Kováč

prof. Dr. Ing. Zbyněk Raida

předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

LICENČNÍ SMLOUVA

POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Bc. Vojtěch Skládáný
Bytem: Bernartice 280, 790 57
Narozen/a (datum a místo): 11. května 1984 v Jeseníku
(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií
se sídlem Údolní 53, Brno, 602 00
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
prof. Dr. Ing. Zbyněk Raida, předseda rady oboru Elektronika a sdělovací technika
(dále jen „nabyvatel“)

Čl. 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- ☐ disertační práce
 - ☒ diplomová práce
 - ☐ bakalářská práce
 - ☐ jiná práce, jejíž druh je specifikován jako
- (dále jen VŠKP nebo dílo)

Název VŠKP: Technologie IEEE802.15.4, ZigBee a příklady jejich aplikací

Vedoucí/ školitel VŠKP: Ing. Michal Kováč

Ústav: Ústav radioelektroniky

Datum obhajoby VŠKP: _____

VŠKP odevzdal autor nabyvateli*:

☒ v tištěné formě – počet exemplářů: 2
☒ v elektronické formě – počet exemplářů: 2

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

* hodící se zaškrtněte

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: 30. května 2008

.....
Nabyvatel

.....
Autor

Abstrakt

Diplomová práce se zabývá vytvořením aplikace a realizace nástroje pro projektování a instalaci radiových sítí podle IEEE802.15.4. Je zaměřena na konstrukci vysílacího a přijímacího zařízení z hlediska mobility a vizualizace naměřených dat v terénu. Shrnuje a popisuje programové vybavení zajišťující měření a ovládání příslušných periférií. Systém je napsán v jazyce C a využívá knihoven MeshNetics OpenMAC. Poslední část práce je zaměřena na praktické měření a testování nástroje pro projektování a instalaci radiových sítí.

Klíčová slova

Technologie IEEE802.15.4., ZigBit, MeshNetics, OpenMAC, XBee, MaxStream, ZigBee

Abstract

The main goals of diploma thesis are application and realization of tool for projection and installation wireless networks according to IEEE802.15.4. The thesis is specialized in construction of transmitter and receiver device in term of mobility and visualization of data measured in terrain. Than the software support for measuring and control appropriate periphery are summarised and described. The system is defined in C programming language and uses OpenMAC library. Last part of thesis is focus on practical measurement and testing tool for projection and installation wireless network.

Keywords

Technologies IEEE802.15.4., ZigBit, MeshNetics, OpenMAC, XBee, MaxStream, ZigBee

Bibliografická citace

SKLÁDANÝ, V. Technologie IEEE802.15.4, ZigBee a příklady jejích aplikací. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 57 s. Vedoucí diplomové práce Ing. Michal Kováč.

Prohlášení

Prohlašuji, že svou diplomovou práci na téma Technologie IEEE802.15.4, ZigBee a příklady jejich aplikací jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 30. května 2008

.....
podpis autora

Poděkování

Děkuji panu Jiřímu Gutmanovi vedoucímu projektu ve firmě BetaControl, Ing. Jindřichu Jeřábkově technickému řediteli firmy BetaControl a vedoucímu diplomové práce Ing. Michalu Kováčovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne 30. května 2008

.....
podpis autora

Obsah

Úvod.....	1
1 Technologie ZigBee	1
1.1 Seznámení s ZigBee	1
1.2 Fyzická a MAC vrstva standardu IEEE802.15.4.....	2
1.3 Struktura komunikačního standardu	4
1.4 Topologie sítě, zabezpečení	4
1.5 Využití ZigBee	6
2 ZigBit modul.....	7
3 Xbee/Xbee PRO Modul	9
4 Řídící Mikroprocesor	11
5 Zobrazovací zařízení	12
5.1 8-bitová komunikace	13
5.2 4-bitová komunikace	13
5.3 Znaková sada HD44780.....	13
5.4 Vytvoření vlastního znaku	14
5.4.1 Zobrazení znaků	15
5.5 Seznam instrukcí HD44780	16
5.6 Technické parametry PC 1602-L.....	16
6 Aplikace Range Measurement Tool.....	17
6.1 Transmitter	17
6.1.1 Program Transmitter.....	17
6.2 Receiver	19
6.2.1 Program Receiver.....	19
6.3 OpenMAC.....	21
7 Řídící program	22
7.1 Inicializace MCU	22
7.1.1 Inicializace SCI.....	23
7.2 Ovládání HD44780 (LCD1602).....	24
7.2.1 Inicializace LCD	24
7.2.2 Vypisování textu	26
7.2.3 Potvrzování dat pro LCD (pin Enable).....	26
7.2.4 Odřádkování.....	27
7.3 Zpracování naměřených dat	27
7.3.1 Indikace ztráty signálu.....	29
7.3.2 Volba přijímacího modulu	29
7.3.3 Přerušení tlačítek.....	30
7.3.4 Menu.....	31
7.3.5 Funkce main.....	33
8 Konstrukce	34
8.1 Vysílací zařízení.....	34
8.1.1 Schéma vysílače.....	34
8.1.2 DPS vysílače a osazení	36

8.2 Přijímací zařízení.....	37
8.2.1 Schéma přijímače	37
8.2.2 DPS přijímače a osazení	39
8.2.3 Generování záporného napětí na Vee.....	41
8.3 Zařízení v praxi	43
8.3.1 Vysílací zařízení.....	43
8.3.2 Přijímací zařízení	44
9 Měření v praxi	48
9.1 Odběr elektrické energie	48
9.1.1 Přijímací zařízení	48
9.1.2 Vysílací zařízení.....	49
9.2 Shrnutí	50
Závěr.....	52
Literatura	54
Seznam symbolů	55
Seznam zkratk	56
Přílohy	57

Úvod

Jako aplikaci realizují nástroj pro projektování a instalaci radiových sítí podle standardu IEEE802.15.4. Koncepte bude sestavena z vysílače osazeným modulem ZigBit od firmy MeshNetics. Vysílač bude bateriově napájen a bude trvale posílat zprávy. Přijímač vybavený řádkovým LCD displejem obsluhovaný pomocí přidaného mikroprocesoru bude zpracovávat zprávy z modulu MaxStream a MeshNetics. Displej bude zobrazovat parametry přijímaného signálu, jako jsou síla signálu, kvalita linky, počet bitových, rámcových chyb a číslo přijatého rámce. Výběr zobrazovaného parametru na LCD bude ovládán pomocí tlačítek umístěných na přijímacím zařízení. Napájení přijímacího zařízení může být zajištěno jak z externího zdroje, tak i z baterií, které představují možnost mobilního využití přijímače a stejně tak vysílače. Komunikace přijímače s okolím je zabezpečena sériovou linkou RS232, která odesílá zprávy přijaté z modulů MaxStream nebo MeshNetics.

1 Technologie ZigBee



1.1 Seznámení s ZigBee

Tato technologie bezdrátové komunikace označená jako ZigBee je vytvořena, aby doplnila trh o zařízení, které může nabídnout:

- velmi nízkou cenu
- nízký příkon při obousměrné komunikaci
- bezdrátové telekomunikační standardy založené na IEEE802.15.4 standardech

Využití ZigBee v bezdrátové komunikaci je možné zařadit do kategorie bezdrátové ovládání a řízení v automatizační technice. V dnešní době existují asi čtyři základní bezdrátové technologie:



GSM (Global System for Mobile Communication) ⇒ Mobilní telefony



Wireless LAN (WLAN) ⇒ připojení přenosných zařízení na lokální síť




Bluetooth ⇒ osobních sítí (PAN), pro komunikaci např. PDA, headset



ZigBee ⇒ ovládání a řízení v automatizační technice

Jednotlivé standardy jsou rozdílné a liší se jak rychlostí přenosu dat, tak i možností dosažitelné maximální vzdálenosti mezi jednotlivými zařízeními potřebnou k úspěšnému navázání spojení. Důležité parametry jsou zobrazené v Tab. 1.1: Srovnání standardů. ZigBee postavené na standardu 802.15.4 dosahuje maximální přenosové rychlosti až 250 Kb/s. Tato přenosová rychlost nám bez problému zaručí možnost využití ZigBee pro ovládání a sledování ovládaných zařízení, které pomocí této technologie propojíme. Díky použití multiskokového ad-hoc směrování umožňuje komunikaci i na větší vzdálenosti bez přímé radiové viditelnosti jednotlivých zařízení. Primární určením směřuje do aplikací v průmyslu a senzorových sítích. Maximální vzdálenost komunikace mezi jednotlivými

moduly je 100 m a více. Například OEM RF Modul XBee-Pro od firmy  **MaxStream®** umožňuje komunikaci ve volném prostředí až na vzdálenost 1200 m. Pomocí takových vzdáleností lze bez problému propojovat jednotlivé budovy velkých firem či dosahovat velkých vzdáleností v uzavřených prostorech pro navázání stabilního spojení [10].

Tab. 1.1: Srovnání standardů



Označení	GSM/GPRS	Wi-Fi™	Bluetooth™	ZigBee™
Standard	1xRTT/CDMA	802.11b	802.15.1	802.15.4
Zaměřeno na	Široké oblasti Hlas & Data	Web, Email, Video	Cable Replacement	Monitoring & Control
Velikost firmwaru	16 MB a více	1 MB+	250 KB+	25 KB – 50 KB
Doba života baterie (dny)	1 – 7	0.5 – 5	1 - 7	100 - 1000+
Počet zařízení v jednom segmentu	1	32	7	255 / 65,000
Přenosová rychlost (Kb/s)	64 - 128	11,000+	720	20 - 250
Dosah zařízení (m)	1000 i více	1 – 100	1 - 10+	1 - 100+
Vyhledání nového zařízení	-----	-----	do 3 s	typicky 30 ms
Výhody	Dosažitelnost, Kvalita	Rychlost, Flexibilita	Cena, Vhodnost	Spolehlivost, Cena

1.2 Fyzická a MAC vrstva standardu IEEE802.15.4

Standard IEEE802.15.4 definuje několik základních radiových pásem, aby mohl být využit v různých zemích, kde jsou rozdílné národní předpisy a normy. Hlavním problémem u většiny bezdrátových technologií jsou rozdílné definice radiových pásem v Americe a v Evropě. Z tohoto důvodu byli definovány tři základní frekvenční pásma: globální, Amerika a Evropa.

Všechny pásma obsahují celkem 26 kanálů, které pracují v bezlicenčních pásmech. Přenosová rychlost činí 20, 40, 250 Kbit/s. První pásmo je na frekvenci 868.3 MHz a obsahuje Channel 0 používané v Evropě. V pásmu 902 MHz – 928 MHz pro Ameriku leží celkem 10 kanálů (Channel 1 - 10). Šířka každého kanálu je 2 MHz a použití BPSK modulace. Jako celosvětově používané pásmo je na frekvenci 2,4 GHz s 16 kanály modulované O-QPSK a šířkou kanálu 5 MHz. Pro přehlednější rozdělení pásem viz Tab. 1.2: Frekvence. MAC vrstva standardu je:

- Postavená na IEEE802.15.4 D18 Protokol
- Schopna zajistit obousměrné připojení v half-duplex režimu
- Podpora pro nízké latence mezi zařízením
- CSMA-CA přístup kanálu
- AES-128 zabezpečení
- Detekce nosné vlny vícenásobný přístup – kolizní vyvarování se

<i>Tab. 1.2: Frekvence</i>	BAND	pokrytí	přenosová rychlost	Kanály
2,4 GHz	ISM	Celosvětové	250Kbps	16
868MHz	ISM	Evropa	20Kbps	1
915MHz	ISM	Amerika	40Kbps	10

Pro dálkové monitorování a ovládání síťových aplikací je zapotřebí jedinečná adresa, která zajistí rozlišení každého modulu v síti. ZigBee umožní rozmístit jednotlivé moduly bezdrátové sítě za nízkou cenu, s nízkým odběrem a s dlouhou životností na bateriový provoz. Lokální adresování v jednoduchých sítích je možno použít pro více než 65,000 (2^{16}) uzlů, které mohou být konfigurovány s redukovanou horní adresou.

U zabezpečovací úrovně najdeme tři stupně:

- Bez zabezpečení
- Access control list
- Symmetric key employing AES-128

MAC vrstva (linková) definuje samotný komunikační protokol, jež je založen na přenosu datových rámců. Jsou definovány čtyři typy komunikačních rámců, které jsou využívány pro přenos dat, řízení či správu sítě:

- **Data Frame** – využíváný pro všechny přenosy užitečných dat
- **Acknowledgement Frame** – rámeček pro přenos potvrzovací informace, využíván na úrovni MAC pro potvrzování komunikace
- **Beacon Frame** – rámeček používaný koordinátorem k vysílání tzv. beacons (používané pro uvádění klientských zařízení do spánkového režimu)
- **MAC Command Frame** – rámeček k nastavování a řízení klientských zařízení v síti ZigBee

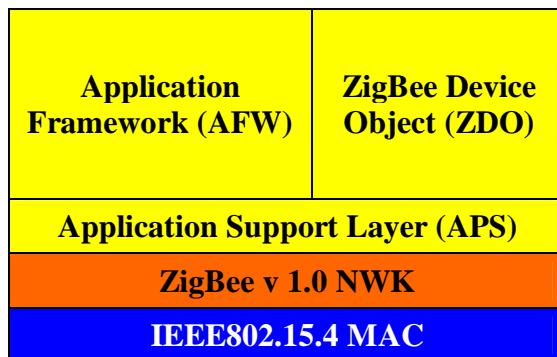
Z důvodů minimalizace spotřeby koncových zařízení mohou být na základě synchronizace mezi koordinátorem sítě a koncovou stanicí uspávána jednotlivá koncová zařízení. K jejich probouzení dochází v předem definovanou dobu a poté jsou přeneseny veškeré užitečné informace. Interval

synchronizačních sekvencí může probíhat v rozmezí 15 ms až přibližně 15 minut. Synchronizace je realizována pomocí tzv. **rámce beacon** (Beacon Frame). Koncová zařízení jsou periodicky probouzena a přenáší data ke koordinátoru sítě. Ten tato data ukládá a následně přeposílá při probuzení zařízení, pro něž jsou tato data určena. Tento přístup umožňuje extrémní snížení spotřeby koncových zařízení. Koncová zařízení tak mohou být napájena bateriově. Při využití všech úsporných opatření je možné dosáhnout výdrže koncového zařízení na jednu alkalickou baterii 6 měsíců až 2 roky. Pokud síť funguje bez použití beacon sekvencí, jednotlivá zařízení periodicky dotazují koordinátora [10].

- **Non-beacon network** – jednotlivá zařízení dotazují koordinátora
- **Beacon enable network** – koncová zařízení jsou po vykonání spojení s koordinátorem uspána.

1.3 Struktura komunikačního standardu

Z důvodu nutnosti implementovat standard ZigBee i do málo výkonných 8 bitových mikroprocesorů (HC08) bylo dbáno na maximální jednoduchost implementace protokolů. Díky tomu struktura protokolů nezabere více než 30 kB programové paměti. Protokol se skládá z tří základních vrstev. Vrstvy standardu IEEE802.15.4, nad nimiž je definována síťová vrstva (NWK) a aplikační vrstva (APL). Fyzická vrstva specifikuje přístup k přenosovému médium. Síťová vrstva realizuje připojení k síti, zabezpečení a směrování paketů. Aplikační vrstva (APL) zajišťuje potřebné služby. Skládá se z aplikační podvrstvy (APS), ZigBee objektů a uživatelských aplikačních objektů.



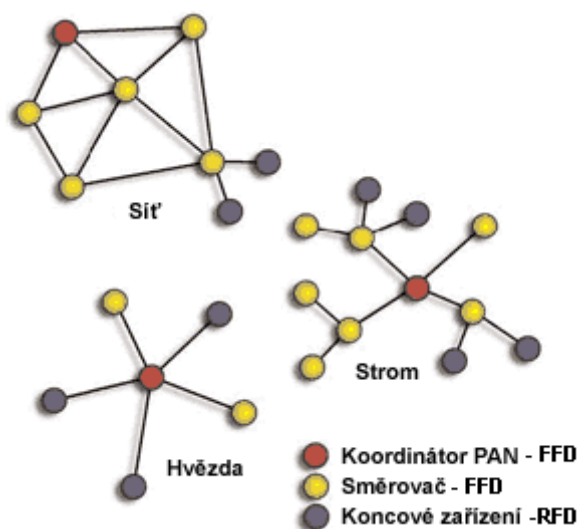
Obr. 1.1: Diagram ZigBeeNet

Na Obr. 1.1: Diagram ZigBeeNet je zobrazen diagram, pro jehož vrstvy jsou charakteristické následující klíčové rysy: síťová nebo stromová topologie, podpora pro velké množství uzlů, volitelný GTS (Guaranteed Time Slot) a zabezpečení, jednoduché API (Application Programming Interface), optimalizace pro nízkou spotřebu, Channel/networks scan selection [10].

1.4 Topologie sítě, zabezpečení

Technologie ZigBee postavená na fyzické linkové vrstvě IEEE802.15.4 definuje tři různé síťové topologie. Základní topologií je topologie hvězdicová s centrálním řídicím uzlem (koordinátorem sítě). Druhým typem je stromová struktura, jež umožňuje zvětšit vzdálenost mezi koordinátorem a

koncovým zařízením. Protokol též umožňuje vytvoření redundančních spojení a vyniká tak topologie typu sítě – mesh. S její pomocí je možné vytvořit síť prakticky libovolného uspořádání.



Obr. 1.2: Topologie sítě ZigBee

Standard ZigBee dělí zařízení na zařízení FFD (Full Functional Device) a RFD (Reduced Functionality Device). FFD zařízení implementují kompletní protokolový rámec a zajišťují veškeré služby, které standard ZigBee stanovuje. RFD zařízení implementují pouze nezbytné protokolové knihovny z důvodu maximálního omezení hardwarové náročnosti. RFD zařízení mohou pracovat pouze jako koncová, mohou komunikovat pouze s koordinátorem sítě a jsou omezeny na hvězdicové uspořádání topologie (koncové větve). Koordinátor sítě a směrovače jsou realizovány FFD zařízeními.

Jednotlivá zařízení sítě jsou adresována pomocí binárního adresního kódu o délce 64 bitů, či ve zkrácené podobě 16 bitů. Lokální zkrácená adresa umožňuje v jedné síti adresovat maximálně 65535 zařízení. Každá sestavená síť je dále identifikována 16 bitovým PAN ID, jež slouží pro rozlišení překrývajících se sítí postavených na standardu IEEE802.15.4. Každou síť zakládá a spravuje koordinátor, který také přiděluje PAN ID. Ostatní stanice pracují jako směrovače, nebo koncová zařízení.

Síťová vrstva (NWK) zajišťuje připojení k sítí, zabezpečení, směrování a synchronizaci. V případě koordinátora sítě je také zodpovědná za start sítě a přiřazování adres nově nalezeným zařízením.

Jako základní zabezpečení ZigBee se používá AES (Advanced Encryption Standard) s klíčem o délce 128 bitů, který je implementován v síťové vrstvě. Pokud je požadováno i zabezpečení MAC Command Frame, Beacon Frame a Acknowledgement Frame je realizováno již v MAC vrstvě pomocí AES. Díky tomu je možné ověřit autenticitu a integritu MAC rámce a zajistit jeho důvěrnost. Při požadavku na ověření integrity je vytvořen MIC (Message Integrity Code) o délce 4, 8 či 16 oktetů a je vyslán společně s MAC rámcem. V tomto případě je použit AES algoritmus v CTR (Counter) módu. Pokud je nutné zajistit důvěrnost MAC rámce, je k němu přidána informace o pořadí rámce a klíče (Frame Count, Key Sequence Count). Na vysílací a přijímací straně je udržována aktuální informace o čísle rámce. Pokud obdrží přijímací zařízení rámec s neplatným číslem je detekováno narušení bezpečnosti. AES je použit v CBC-MAC (Cipher Block Chaining) módu. AES se užívá také v tzv. módu CCM a to při ověřování integrity sítě a při šifrování dat.

Síťová vrstva využívá k zabezpečení SSP (Security Services Provider). Tato vrstva zajišťuje zabezpečení odchozích rámců, dekodování rámců a ověřování pravosti příchozích rámců. Jako zabezpečovací algoritmus je použit AES v mírně modifikovaném módu jež nese označení CCM. Vyšší vrstvy se starají o nastavení SSP, určují tak nastavení klíčů a udávají jakým způsobem bude použit CCM pro jednotlivé rámce [10].

1.5 Využití ZigBee

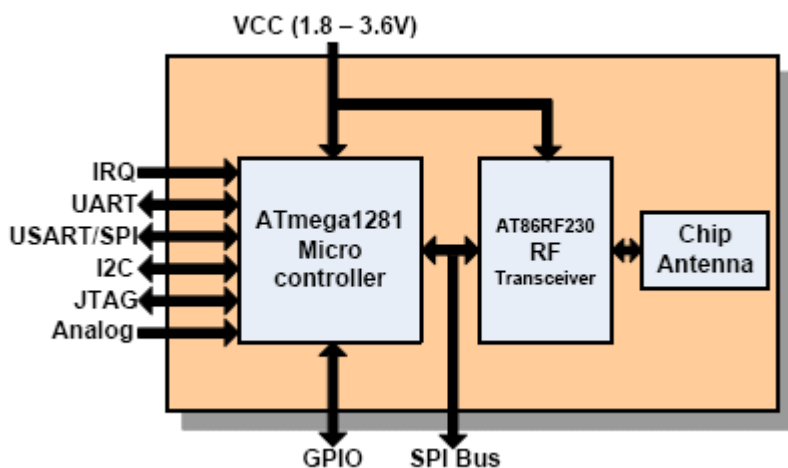
Bezdrátové technologie lze využít v běžné každodenní praxi. Hlavní výhodou v aplikaci bezdrátové technologie je možnost řízení připojeného zařízení při volném pohybu koncového ovládacího zařízení a zároveň také odpadá nutnost instalace spotřebního materiálu jako jsou např. dráty, a práce s tím spojena. Jednou z nejpoužívanějších aplikací každodenního života je možnost ovládat pomocí ZigBee TV, VCR, DVD/CD, radia, atd. Další možností využití je komunikace mezi ovládacími prvky, mezi které patří např. myš k PC, klávesnice. Jako další využití bezdrátových technologií lze jmenovat zabezpečení přístupu do budov, ovládání světel, monitorování životních funkcí člověka pohybujícího se v dané budově, možnost řízení procesů spotřeby energie v budovách apod. Tato technologie je schopna posloužit jak jedinci, tak zajistit automatizaci budov či průmyslové výroby. Níže jsou v bodovém výčtu uvedeny možnosti využití:

- Kontrolní monitor
- Monitorování : teplota, tlak, tok, vlhkost, chvění, atd.
- Automatický odečet elektroměrů
- Regulace osvětlení
- Bezdrátový detektor kouře a CO detektory
- Regulace vytápění
- HVAC kontrola
- Řízení zásob
- Environmentální ovládací prvky
- Bezpečnost

2 ZigBit modul

Jednou ze součástí využitých při konstrukci přijímacího zařízení je modul ZigBit s dvojitou čipovou anténou. Jedná se o nízkopříkonový, best-in-class sensitivity 802.15.4 / ZigBee module s velice malými rozměry 13.5 x 24 mm sestavený s populární ATmega 1281 a posledního AT86RF230 rádiového transceiveru. ZigBit modul s dvojitou čipovou anténou odstraňuje nutnost drahého a náročného RF vývoje a zkracuje tak dobu potřebnou pro okamžité použití. Použitý 802.15.4/ ZigBee software zabezpečí standard založený na bezdrátovém připojení pro širokou škálu využití.

ZigBit je založený na pokrokové hardwarové platformě Atmel 802.15.4. Výkonný mikroprocesor ATmega1281 disponuje 128 kB flash a 8 kB RAM paměti. Integrovaný AT86RF230 transceiver nabízí citlivost od -101 dBm pro Rx příjem a až +3 dBm pro vysílací výkon [8].



Obr. 2.1: ZDM-A1281-A2 Blokový Diagram

Tab. 2.1: Vlastnosti ZigBit OEM Modulu

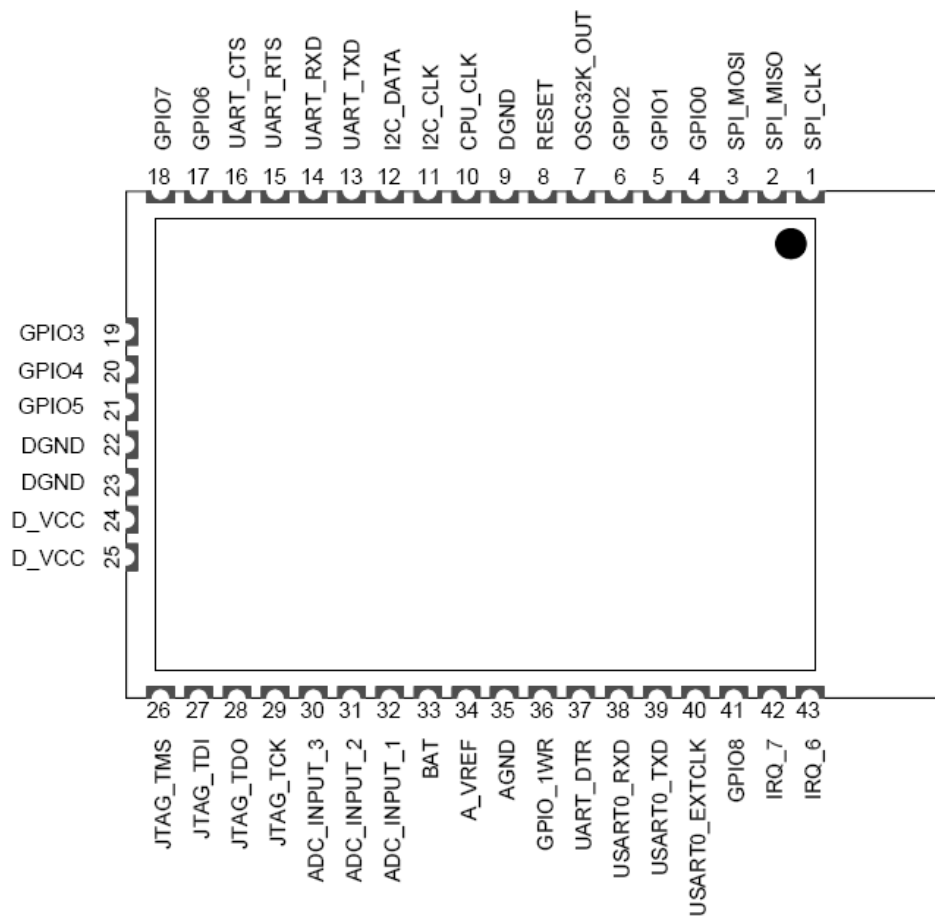
- Malé rozměry: 13.5 mm x 24 mm
- Dvojitá čipová anténa
- Velmi malá energetická spotřeba (6 μ A v režimu spánku)
- Vysoká imunita proti rádiovému rušení
- Široký okruh podporovaných rozhraní s referenčním softwarovými ovladači

Tab. 2.2: Specifikace ZigBit RF Modulu

Parameter	ZigBit
Frequency band	2.400 - 2.483 GHz
Number of channels	16
Data rate	250 kbps
Max output power	3 dBm
2nd harmonic	-28 dBm
3rd harmonic	-26 dBm
Sensitivity (PER 1%)	-101 dBm
Adjacent Channel Rejection	27 dB
Alternate Channel Rejection	53 dB
Supply voltage	1.8 – 3.6 V
Current consumption, RX	19 mA
Current consumption, TX	18 mA
Current consumption (Sleep mode)	6 μ A
Flash memory	128 kB
RAM	8 kB
EEPROM	4 kB
Operating Temperature	-40 — +85 °C

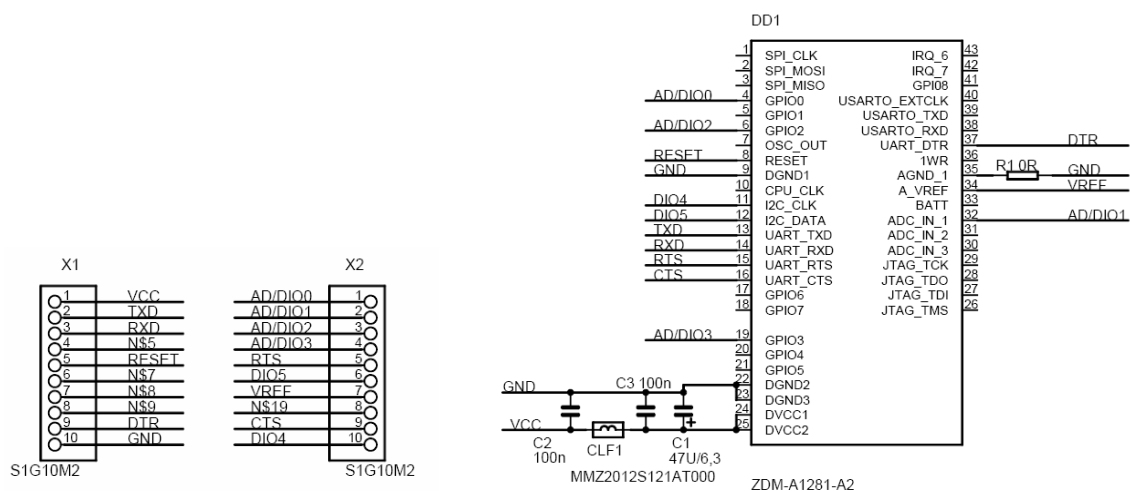
Tab. 2.3: Podporovaná externí rozhraní

- USART/SPI, I2C, 1-wire
- UART with CTS/RTS control
- JTAG
- 9 spare GPIOs (up to 25 GPIOs total)
- 2 spare IRQ lines
- 4 ADC lines



Obr. 2.2: Popis pinů ZigBit (SMD)

Na Obr. 2.2 je znázorněno rozmístění pinů na ZigBit, ale v přijímacím zařízení bude využita stejná patice jakou má XBee. Proto za pomoci redukce vytvořené firmou BetaControl, získáme rozložení vývodů podobné jako u XBee viz Obr. 2.3.



Obr. 2.3: a) Rozmístění vývodů na redukcí ZigBit → XBee b) Schéma připojení ZigBit k pinům

3 Xbee/Xbee PRO Modul



Dalším použitým modulem pro příjem a měření kvality signálu je modul Xbee. Tento modul byl zkonstruován pro práci se standardem ZigBee / IEEE802.15.4. a zaplnil tak místo v levných, nízkopříkonových zařízeních v bezdrátových sítích. Moduly s minimálním příkonem poskytují spolehlivý přenos dat mezi jednotlivými zařízeními i při velmi nízké kvalitě signálu, pracující v ISM pásmu na frekvenci 2,4 GHz. Verze Xbee a Xbee-PRO jsou mezi sebou navzájem kompatibilní.

Nabízejí vysokou výkonnost s možností uskutečnění spolehlivého spojení až na vzdálenost 30 m pro Xbee a 100 m u Xbee-PRO ve vnitřních prostorech. Příjemná citlivost Xbee modulu je -92 dBm s RX proudem 50 mA. Rozsah napájecího napětí se může pohybovat v rozmezí 2,8 – 3,4 V.

Tab. 3.1: Výkonnost Xbee/XBee-PRO

Xbee	XBee-PRO
<ul style="list-style-type: none"> Vnitřní prostory/Město: až 30 m Venkovní prost., přímá viditelnost: až 100 m Výkon vysílače: 1 mW (0 dBm) Citlivost přijímače: -92 dBm 	<ul style="list-style-type: none"> Vnitřní prostory/Město: až 100 m Venkovní prost., přímá viditelnost: až 300 m Výkon vysílače: 100 mW (20 dBm) EIRP Citlivost přijímače: -100 dBm

RF rychlost přenosu: 250,000 bps

Tab. 3.2: Malá spotřeba

XBee	XBee-PRO
TX proud: 45 mA (@3.3 V)	TX proud: 270 mA (@3.3 V)
RX proud: 50 mA (@3.3 V)	RX proud: 55 mA (@3.3 V)
Režim spánku: < 10 μ A	Režim spánku: < 10 μ A

XBee (top view)	XBee-PRO (top view)	XBee & XBee-PRO (side views)

Obr. 3.1: Rozměry a popis XBee/XBee-PRO

Naprogramovaný modul XBee pro měření kvality signálu není potřeba nijak řídit a lze jej zapojit s minimální doporučenou konfigurací **à** Vcc, RESET, GND, DOUT. Takto osazený modul přijímá data přenášená v prostředí a po zpracování dat pošle výsledek po DOUT (chová se jak Tx_UART) s nastavenou rychlostí 38400 bps, s 8-bit datovým modem, bez parity a jedním stop bitem. Popis jednotlivých pinů XBee modulu je v Tab. 3.3 viz technické dokumentace [17].

Tab. 3.3: Popis pinů XBee/XBee-PRO

Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN / CONFIG	Input	UART Data In
4	CD / DOUT_EN / DO8	Output	Carrier Detect, TX_enable or Digital Output 8
5	RESET	Input	Module Reset
6	PWM0 / RSSI	Output	PWM Output 0 or RX Signal Strength Indicator
7	[reserved]	-	Do not connect
8	[reserved]	-	Do not connect
9	DTR / SLEEP_RQ / DI8	Input	Pin Sleep Control Line or Digital Input 8
10	GND	-	Ground
11	AD4 / DIO4 / RF_TX	Either	Analog Input 4, Digital I/O 4 or Transmission Indicator
12	DIO7 / CTS	Either	Digital I/O 7 or Clear-to-Send Flow Control
13	ON / SLEEP	Output	Module Status Indicator
14	VREF	Input	Voltage Reference for A/D Inputs
15	AD5 / DIO5 / Associate	Either	Analog Input 5, Digital I/O 5 or Associated Indicator
16	AD6 / DIO6 / RTS	Either	Analog Input 6, Digital I/O 6 or Request-to-Send Flow Control
17	AD3 / DIO3 / COORD_SEL	Either	Analog Input 3, Digital I/O 3 or Coordinator Select
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0	Either	Analog Input 0 or Digital I/O 0

4 Řídící Mikroprocesor

Jako řídicí mikroprocesor, který bude zajišťovat komunikaci s přijímacími moduly jsem vybral mikroprocesor MC9S08GB60. Jedná se o verzi vyráběnou v pouzdře s 64 – piny a označením LQFP. Obsahuje dvě sériové linky, které nám zajistí možnost přijímání dat ze dvou různých zdrojů. V našem případě bude jeden zdroj ZigBit a druhý zdroj XBee/PC. Mikroprocesor rodiny HCS08 disponuje 60kB FLASH pamětí, 4kB RAM a poskytuje 56 univerzálních vstup/výstupních piny. Dále je vybaven jedním tří kanálovým a jedním pěti kanálovým šestnácti bitovým časovačem. Nabízí také možnost využít až 32 přerušení, kde pro řízení a ovládání přijímacího zařízení využijeme pouze tři přerušení. Nejvíce využívané přerušení je sériové linky SCI1 a SCI2, které nám přijímají data z přijímacích modulů. Jako třetí přerušení využijeme reakci na stisk tlačítka. Mikroprocesor obsahuje celkem sedm vstup/výstupních portů, kde využijeme port A pro připojení ovládacích tlačítek, port B k připojení zobrazovacího zařízení a port D pro připojení LED, které budou zobrazovat stav a činnost mikroprocesoru.

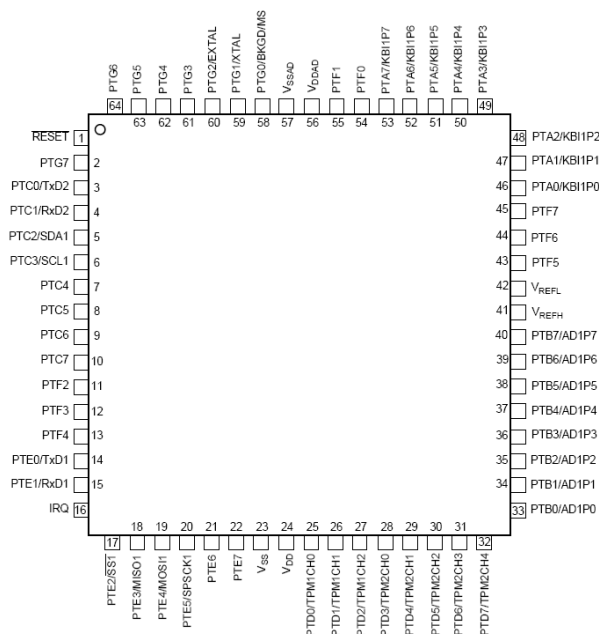
Tab. 4.1: Stručný přehled MC9S08GB60

Zařízení	FLASH	RAM	Časovače	Portů	Sériová linka	SPI	Hodiny	A/D převodník
MC9S08GB60	60kB	4kB	jeden 3-kanálový a jeden 5 kanálový 16-bitový časovač	7 (A-F)	SCI1 a SCI2	1	Interní / Externí	8-kanálový 10-bitový

Tab. 4.2: Popis zapojených vývodů mikroprocesoru

Port	Piny	Popis připojení
A	PTA2-PTA5	Ovládací tlačítka
B	PTB0-PTB3	Datová sběrnice LCD
D	PTD6-PTD7	Řídící sběrnice LCD
	PTD0-PTD3	Informační LED diody
C	PTC0-PTC1	SCI2 sériová komunikace se ZigBit
E	PTE0-PTE1	SCI1 sériová komunikace s XBee/PC
G	PTG0	Programovací port BDM

Obr. 4.1: Rozmístění vývodů



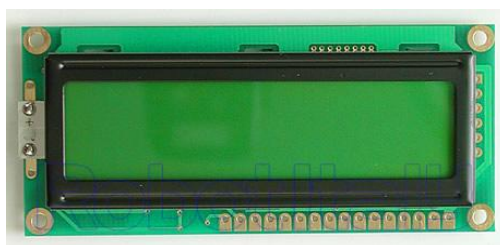
Tab. 4.3: Využité vektory přerušení

Číslo vektoru	Adresa (Horní/Dolní)	Jméno vektoru	Modul	Zdroj	Spuštění (Aktivace)	Popis
22	\$FFD2/FFD3	Vkeyboard1	KBI	KBF	KBIE	Keyboard pins
20	\$FFD6/FFD7	Vsci2rx	SCI2	IDLE RDRF	ILIE RIE	SCI2 receiver
17	\$FFDC/FFDD	VSCI1rx	SCI1	OR NF FE PF	ORIE NFIE FEIE PFIE	SCI1 receiver

5 Zobrazovací zařízení

Jedno z nejefektivnějších a nejúčinnějších řešení zobrazení naměřených výsledků je využití LCD displeje, který efektně a jasnou formou sdělí pozorovateli přesnou hodnotu i s případným komentářem. Výhoda LCD displejů je také ta, že si lze zvolit dvě možnosti připojení displeje k mikroprocesoru. Můžeme si vybrat 8-bitové datové připojení nebo 4-bitové datové připojení, tento výběr by mohl vést k připojení více zařízení na jednu bránu, nebo v případě nutnosti napojení malého počtu vývodů zajistit redukci vodičů na minimum.

LCD-displeje se vyrábějí v mnoha podobách a verzích. V ČR je možné na trhu v ČR sehnat snad jakýkoli druh. LCD-displeje se vyrábějí jak textové tak grafické. Textové nabízejí různé možnosti zobrazení od jednořádkových s osmi znaky na řádek až po čtyřřádkové s možností čtyřiceti znaků na řádek. Grafické LCD-displeje jsou s různým rozlišením. Pro měření kvality signálu na přijímacím zařízení jsem zvolil textové LCD a to jak z důvodu jednoduššího ovládání, tak z důvodu ekonomické úspory. Pravděpodobně nejvhodnějším LCD-displejem je verze 16x2 (16 sloupců a 2 řádky) s řadičem HD44780. Fotografie je přiloženo na Obr. 5.1 LCD XIAMEN OCULAR a s označením GDM1603K. V přijímacím zařízení je použito LCD od firmy POWERTIP verze PC 1602-L. Toto LCD je rozměrově a vzhledově totožné, ale liší se v možnosti pracovat s napětím od 2,7 V [12].



Obr. 5.1: LCD XIAMEN OCULAR - GDM1603K

Tab. 5.1: XIAMEN OCULAR - GDM1603K	
LCD Pin Number	LCD Pin Description
6	Enable
7	Data 0
8	Data 1
9	Data 2
10	Data 3
11	Data 4
12	Data 5
13	Data 6
14	Data 7
5	Read/Write
4	RS
1	Ground
2	+5V

5.1 8-bitová komunikace

Při tomto druhu komunikace je využito všech osmi datových vodičů DB0-DB7. Pomocí této komunikace dochází k přivedení dat na datovou sběrnici a následné potvrzení signálem na E-enable. Tento způsob jsem ověřil. Komunikace s LCD byla bez problémů. 8-bitová komunikace je jednoduchá a výkonná z hlediska zaneprázdnění mikroprocesoru [12].

5.2 4-bitová komunikace

Při tomto druhu komunikace jsou využity jen 4-bity datové sběrnice DB4-DB7. Vodiče DB0-DB3 jsou uzemněny. Komunikace funguje naprosto stejně jako při 8-bitové komunikaci, jen data jsou vyslána nadvakrát. Při inicializaci displeje je zapotřebí první instrukci poslat jen jako první polovinu instrukce a poté potvrdit E-enable. Následně se instrukce opakuje, ale jsou poslány obě poloviny inicializace s povolením nastavení dvou řádkového displeje a rozměry fontu.

Dvojitě poslání horní poloviny inicializačního kódu je nutné, neboť uvede LCD do režimu 4-bit datové komunikace. Pouhé poslání inicializační instrukce na dvakrát je nedostačující a LCD nebude komunikovat ve 4-bit režimu!

Jedna z velkých nevýhod při zapojení LCD 4-bit datové komunikace s mikroprocesorem je nutnost použití dalších tří vodičů k řízení komunikace. V tomto projektu využívám zapojení LCD pomocí 4 datových vodičů DB4-DB7, poté jeden vodič pro potvrzení dat (E-enable) a jeden vodič na rozlišení instrukcí a dat. Vodič připojený na pin 5(Read/Write) potřebný k určení čtení nebo zápisu je uzemněn a LCD je uveden pouze do režimu zápisu, tím je celkový počet šesti vodičů použitých pro řízení a dále jsou užity dva vodiče pro napájení. U přijímacího zařízení jsem nevyužil možnost podsvícení, i když jí LCD umožňuje. Rozhodl jsem se tak z toho důvodu, že LED diody použité pro podsvícení odebírají proud $I = 320 \text{ mA}$, který je skoro trojnásobný oproti odběru plně osazeného přijímacího zařízení a také při napájecím napětí 4,2 V a s napájecím napětím 3 V by byl jejich účinek v podsvícení zanedbatelný [12].

5.3 Znaková sada HD44780

Použitá zobrazovací jednotka obsahuje řadič HD44780. Ten zabezpečuje veškerou komunikaci mezi procesorem a zobrazenými údaji. Všichni výrobci používají tento typ řadiče nebo jeho ekvivalent, proto nezáleží na výrobci. Pro dosažení plné kompatibility mezi displeji je zapojení vždy stejného přírodního konektoru, což zjednodušuje použití displejů v praxi. Řadiče používají mnoho znakových sad, ale většina displejů obsahuje zde uvedenou znakovou sadu. V sadách se odlišují matice pro abecedy odlišné od latiny (např. azbuka, atd.). Konkrétní sadu obsaženou v daném displeji najdeme v datasheetu výrobce nebo viz Tab. 5.2.

Každý znak z tabulky Tab. 5.2 je zobrazen jako matice 5x8 bodů. Jednotlivé znaky jsou uloženy napevno ve vnitřní paměti ROM. Adresa jednotlivých znaku je složena z horní a dolní poloviny adresy, kterou najdeme na krajích tabulky, kde vertikální sloupce určují horní polovinu a horizontální sloupce určují dolní polovinu adresy. Jak je patrné na adrese 00H až 0FH, není zobrazen žádný znak. Toto je způsobeno tím, že pro dané adresy si lze definovat vlastní znaky. Počet možných definovatelných znaků je osm, ty jsou uloženy na adrese 00H – 07H. Na dalších adresách 08H – 0FH jsou tyto znaky zopakovány. To znamená, že jsou uloženy na dvou místech v paměti. Možnost vytvoření vlastních znaků jsem osobně ověřil [12].

Tab. 5.2: Mapa znaků pro řadič HD44780

	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
xxxx0000	0	0	0	0	1	1	1	1	0	1	0	1	1	0	1	1	1	1	1
xxxx0001	0	0	1	1	0	0	1	1	1	1	0	0	0	1	0	1	1	1	1
xxxx0010	0	0	1	0	1	0	1	1	1	1	0	1	0	0	1	1	0	1	1
xxxx0011	0	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0	1	1
xxxx0100			0	a	P	`	P			-	ア	ミ	α	ρ					
xxxx0101			!	1	A	Q	a	q	。	ア	チ	4	3	Q					
xxxx0110			"	2	B	R	b	r	「	イ	ツ	ズ	β	θ					
xxxx0111			#	3	C	S	c	s	」	ウ	テ	モ	ε	ω					
xxxx1000			\$	4	D	T	d	t	、	エ	ト	ハ	μ	Ω					
xxxx1001			%	5	E	U	e	u	・	オ	ナ	1	5	Ü					
xxxx1010			&	6	F	V	f	v	ヲ	カ	ニ	ヨ	ρ	Σ					
xxxx1011			'	7	G	W	g	w	ア	キ	ズ	ラ	Q	π					
xxxx1100			(8	H	X	h	x	イ	ク	ネ	リ	J	Σ					
xxxx1101)	9	I	Y	i	y	ウ	ケ	ル	リ	U						
xxxx1110			*	:	J	Z	j	z	エ	コ	ハ	レ	i	キ					
xxxx1111			+	;	K	[k	[オ	サ	ヒ	ロ	*	所					
xxxx1100			,	<	L	¥	l	l	ハ	シ	フ	ワ	Φ	所					
xxxx1101			-	=	M]	m	}	ユ	ズ	ヘ	ン	も	÷					
xxxx1110			.	>	N	^	n	→	ヨ	セ	ホ	°	所						
xxxx1111			/	?	O	_	o	←	ッ	ソ	マ	°	ö	■					

5.4 Vytvoření vlastního znaku

Pro vytvoření vlastního znaku nejdříve vyšleme instrukci pro nastavení adresy, kam chceme data zapisovat a to ve formátu kód znaku (0-7), řádek znaku (0-7). Poté vyšleme data sestavená podle viditelnosti bodu. Celý cyklus opakujeme 8krát. Celkem osm řádků pro jeden znak. Daný znak bude potom uložen na zadané adrese a na adrese inkrementované o osm viz Tab. 5.3.

Tab. 5.3: Vytvoření vlastního znaku

Kód znaku	Adresa v CGRAM			Data										
	kód znaku	řádek znaku		7	6	5	4	3	2	1	0			
0, 8	0	0	0	0	0	0	0	0	1	0	1	0		
				0	0	0	1	0	0	1	0	0	0	
				0	1	0	0	0	1	1	1	0	0	
				0	1	1	1	1	0	0	0	1	0	
				1	0	0	x	x	x	1	1	1	1	0
				1	0	1	1	0	0	0	0	0	0	0
				1	1	0	0	1	1	1	0	0	0	0
				1	1	1	0	0	0	0	0	0	0	0
1, 9	0	0	1	0	0	0	0	0	0	0	1	0		
				0	0	0	1	0	0	1	0	0	0	
				0	1	0	0	0	1	1	1	0	0	
				0	1	1	1	0	0	0	0	1	0	
				1	0	0	x	x	x	0	1	1	1	1
				1	0	1	1	0	0	0	0	0	1	0
				1	1	0	0	1	1	1	1	0	0	0
				1	1	1	0	0	0	0	0	0	0	0

5.4.1 Zobrazení znaků

Znaky na displeji lze zobrazit buď ve formátu 5x8 bodů nebo 5x10 bodů. Dále je možné nastavit blikání kurzoru, zobrazit kurzor, posouvání znaků nebo displeje, smazat displej nebo umístit kurzor na požadované místo na displeji. Každý řádek a sloupec na displeji má adresu, aby se dalo k jednotlivým buňkám kdykoliv snadno přistoupit. V Tab. 5.4 jsou vypsány hodnoty adresy buněk.

Tab. 5.4: Hodnoty adres pro dvouřádkové displeje	
Počet znaků	Pozice v DDRAM
2x8	1.řádek 00h .. 07h 2.řádek 40h .. 47h
2x16	1.řádek 00h .. 0Fh 2.řádek 40h .. 4Fh

Některé jednořádkové displeje, 1x16 znaků, se musí inicializovat jako dvouřádkové. Je to zjednodušení a šetření ze strany výrobce, protože jeden řadič LCD displeje umí ovládat 2x8 znaků. Bližší informace najdete v datasheetu každého displeje.

Jak je zmíněno výše, můžeme zapnout posuv displeje. Posun doleva inkrementuje původní adresy. Hodnoty zobrazené v tabulkách jsou správné hodnoty pro LCD 2x16 znaků.

Tab. 5.5: DDRAM bez posuvu displeje

pozice LCD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
adresa DDRAM	0	1	2	3	4	5	6	7	8	9	0A	0B	0C	0D	0E	0F
adresa DDRAM	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

Tab. 5.6 platí jen , když dojde k posunutí o jednu buňku doleva.

Tab. 5.6: DDRAM posun doleva

pozice LCD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
adresa DDRAM	1	2	3	4	5	6	7	8	9	0A	0B	0C	0D	0E	0F	10
adresa DDRAM	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50

Pokud budeme chtít posun doprava, hodnoty v následujících adresách budou viz Tab. 5.7.

Tab. 5.7: DDRAM posun doprava

pozice LCD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
adresa DDRAM	27	0	1	2	3	4	5	6	7	8	9	0A	0B	0C	0D	0E
adresa DDRAM	67	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E

Jak bylo uvedeno v podkapitole 5.4 Vytvoření vlastního znaku, lze definovat vlastní znaky. Maximálně však 8. Není to mnoho a například k doplnění české diakritiky to nestačí, ale ta možnost tu

existuje. Jedna z možných variant jak získat všechny potřebné znaky je pravidelná aktualizace místa v paměti. Bohužel tato možnost nám přepíše i ten znak, který je právě používán. Při takovém řešení musíme dbát na to, aby nám nový znak nepřepisoval právě využívaný znak.

5.5 Seznam instrukcí HD44780

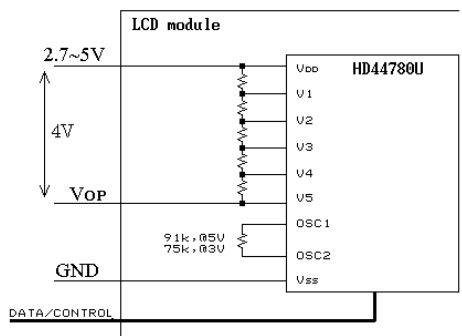
Celý seznam instrukcí je zobrazen v Příloze A. Těmito instrukcemi ovládáme zobrazení na displeji. Dokážeme jak číst, tak zapisovat na zobrazovací jednotce. U každé dané instrukce je zobrazen čas prodlevy, který je potřeba dodržet. Při nedodržení prodlevy dojde k nevykonání následující či případně více následujících instrukcí. Tyto prodlevy označují, že po danou dobu je LCD pro mikroprocesor zaneprázdněný a nebude moci vykonávat žádné námi nově zadané příkazy. Nejdéle prováděné instrukce jsou smazat displej a návrat na pozici 00h [12].

5.6 Technické parametry PC 1602-L

Podle kapitoly 5 Zobrazovací zařízení bylo zapotřebí použít LCD displej, který bude fungovat při napájecím napětí $V_{dd} = 3\text{ V}$ viz Tab. 5.8 [18]. Bohužel pro vytvoření potřebného kontrastu, bylo zapotřebí vytvořit napětí o rozdílu přibližně 4,5 V podle V_{OP} viz Obr. 5.2. Ten to problém je vyřešený v kapitole 8.2.3 Generování záporného napětí na Vee.

Tab. 5.8: Elektrická charakteristika PC 1602-L

PIN ASSIGNMENT			ABSOLUTE MAXIMUM RATING					
Pin no.	Symbol	Function	Item	Symbol	Condition	Min.	Max.	Units
1	Vss	Power supply(GND)	Supply for logic voltage	Vdd-Vss	25°C	-0.3	7	V
2	Vdd	Power supply(+)	LCD driving supply voltage	Vdd-Vee	25°C	-0.3	13	V
3	Vo	Contrast Adjust	Input voltage	Vin	25°C	-0.3	Vdd+0.3	V
ELECTRICAL CHARACTERISTICS								
Item	Symbol	Condition	Min.	Typical	Max.	Units		
Power supply voltage	Vdd-Vss	25°C	2.7	—	5.5	V		
LCD operation voltage	Vop	Top	N	W	N	W	N	W
		-20°C	—	7.1	—	7.5	—	7.9
		0°C	4.5	—	4.5	—	4.7	—
		25°C	4.1	6.1	4.3	6.4	4.5	6.7
		50°C	4	—	4.2	—	4.4	—
		70°C	—	5.7	—	6	—	6.3
LCM current consumption (No B/L)	Idd	Vdd=5V	—	2	3	mA		
Backlight current consumption	LED/edge	VB/L=4.2V	—	120	—	mA		
	LED/array	VB/L=4.2V	—	360	—	mA		



Obr. 5.2: Dělič napětí řízení kontrastu PC 1602-L

6 Aplikace Range Measurement Tool

Vzorová aplikace Range Measurement má za úkol měřit rádiový výkon ZigBit vysílače a srovnat je s platformami dalších výrobců. Tato aplikace využívá navíc knihovnu MAC library (MACLibS.o) obsahující speciální funkce zvláště pro tuto aplikaci. Vzorová aplikace Range Measurement je rozdělena do dvou firmwarových obrazů:

6.1) pro vysílač

6.2) pro přijímač

6.1 Transmitter

Vysílač vykonává následující funkce:

- Generuje pakety každých 6 ms – 10 ms obsahující 1024 bitů ze speciálně vygenerované pseudonáhodné sekvence (podle ITU-T O.151)
- Ovládání RF výstupního výkonu
- Odešle přes RF číslo rámce využitého na UART (v 38400, 8N1 módu)

6.1.1 Program Transmitter

Výčtový typ enum nám umožní vytvořit proměnnou, která může obsahovat hodnoty konstant určených při deklaraci výčtového typu. Proměnná se vytváří pomocí typedef. Deklarace typu enum má hodně společného s deklarací struktury. Lze vytvořit buďto jen výčtový typ enum (stejně jako jenom strukturu), nebo pomocí typedef definovat nový datový typ.

```
enum
{
    BUFFER_DATA_LENGTH = MAX_PHY_PACKET_SIZE + 1, // PSP buffer length.
    NPACKETS             = 10000,                  // number of packets to generate
    MIN_DATA_PERIOD      = 6,
};
```

Za pomoci enum jsou vytvořené proměnné s danými hodnotami konstant jako je počet paketů, nominální doba periody dat a velikost paketů, které budeme přenášet.

```
struct
{
    uint8_t  PSPData[BUFFER_DATA_LENGTH]; // Pseudorandom sequence buffer.
    MACStatus dataReqStatus;               // Status of data request operation.
} dataParam; // Describes parameters for special RangeMeasurementTool data request.
```

Je deklarována i struktura, kde je vytvořený buffer pro pseudonáhodnou sekvenci dat a proměnná popisující aktuální stav dat vyžadované operace.

Program Transmitter je vytvořený pro neustálé vysílání, a proto není zapotřebí vytvářet přijímací funkci pro zpracování dat. Proto je zde vytvořena funkce na odesílání paketů *sendPacket()*.

```
void sendPacket()
{
    if(Npacket2Tx > 0)
    {
        sensTestAuxiliary_sendFrame(dataParam.PSPData, &dataParam.dataReqStatus);
    }
    ...
}
```

Funkce *sendPackets()* je specifikována podmínkou, která respektuje nastavení, potřebné pro odeslání konečné řady deseti tisíc paketů dle nastavení, nebo odesílání nekonečného vysílání dle nastavení. V našem vysílacím zařízení nejsou připojena ovládací tlačítka, a tak je vysílač v režimu neustálého vysílání.

```
void PSPCalculation()
{
    uint8_t i, j;
    uint16_t shiftReg = 0x7418; //just some digits

    for( i = 0; i < BUFFER_DATA_LENGTH; i++ )
    {
        for( j = 0; j < 8; j++ )
        {
            if( PSPNextBit(&shiftReg) )
                dataParam.PSPData[i] = (dataParam.PSPData[i] >> 1) | 0x80;
            else
                dataParam.PSPData[i] = (dataParam.PSPData[i] >> 1);
        }
    }
    dataParam.PSPData[0] = MAX_PHY_PACKET_SIZE;
}
```

Program Transmitter generuje pseudonáhodnou posloupnost dat *PSPCalculation()*. PSP je poté uložena do bufferu *PSPdata*. Vygenerovaná PSP posloupnost dat vysílačem i přijímačem je stejná a na základě ní jsou potom porovnávána přijatá data.

Abychom mohli komunikovat po sériové lince, je zapotřebí její nastavení na požadované parametry uvedené v popisu aplikace. O nastavení sériové linky se postará funkce *appEntry_main()* na požadovaný 38400 8N1 mód. Dále zavolá funkci pro výpočet PSP.

```
void appEntry__main()
{
    UARTMode_t uartMode;

    // UART settings.
    uartMode.baudrate = UART_BAUDRATE_38400;
    uartMode.data = UART_DATA8;
    uartMode.parity = UART_PARITY_NONE;
    uartMode.stopbits = UART_STOPBITS2;
    ...
    // Pseudorandom sequence calculation.
    PSPCalculation();
    // MAC initing.
    MAC_init();
}
```

Vysílání dat provádí funkce *sendPacket()* volající požadovanou funkci, ale funkce, která volá *sendPacket()*, se jmenuje *MLME_STARTconfirm()*.

```
void MLME_STARTconfirm(void)
{
    appTimer_start(timerId, TIMER_REPEAT_MODE, 50);
    changeChannel = FALSE;
    sendPacket();
}
```

Obsahuje funkci volající časovač, který zajistí nastavení požadovaného zpoždění pro odeslání následujícího paketu.

Na vysílacím modulu lze nastavovat i vysílající výkon a to pomocí tlačítek, což zajišťuje funkce *MLME_SETconfirm()*. Jedinou podmínkou je nastavení výkonu před zapnutím ZigBit. Pokud podmínka není dodržena, pak je nutné následně provést RESET.

6.2 Receiver

Přijímač vykonává následující funkce:

- Přijímá pakety z vysílače
- Analyzuje pakety
- Průměruje přijatá data
- Vypočítá kontrolní součty
- Vypočítá statistiky
- Odešle zpracovaná data zahrnující množství bitových chyb, rámcovou chybu a počet přijatých rámců do PC přes UART (v 38400, 8N1 módu)

Přijímaná data můžeme zobrazit na PC při využití Hyper Terminálu nebo nějaké jiné terminálové aplikace, jako například program TeraTerm, který jsem použil pro zachytávání dat na sériové lince. Dále můžeme monitorovat RSSI (Received Signal Strength Indicator) úroveň a LQI (Link Quality Indicator) hodnoty.

6.2.1 Program Receiver

Obdobně jako v programu Transmitter si definujeme výčtový typ, kterým vytvoříme proměnou s konstantami stavu pro obsluhu vysílání po sériové lince.

```
enum
{
    FRAME_COUNTER_OUT_STATE,
    FRAME_ERROR_COUNTER_OUT_STATE,
    BIT_ERROR_COUNTER_OUT_STATE,
    LQI_OUT_STATE,
    RSSI_OUT_STATE,
} uartState; // UART current state.
```

Dále jsou vytvořeny konstanty definující minimální hodnotu RSSI (Received Signal Strength Indicator) a nastavení maximální délky paketu pro vysílání.

```
enum
{
    BAD_FRAMERATE    = 50,           // Minimum rate of the incoming frames.
    MIN_RSSI_VALUE    = 89,           // Minimum RSSI value (described in the
    AT86RF230 datasheet).
    BUFFER_DATA_LENGTH = MAX_PHY_PACKET_SIZE + 1, // PSP buffer length.
};
```

Dále bude obdobně podle stejného algoritmu vypočítána pseudonáhodná sekvence pomocí funkce *PSPCalculation()*, aby bylo možné přicházející data porovnat s identickou pseudonáhodnou posloupností vygenerovanou programem Transmitter a zjistit tak chyby vzniklé na přenosové trase. Další potřebnou funkcí je *appEntry_main()*, která nám nastaví sériovou linku podle požadavků do 38400, 8N1 módu.

Abychom mohli zjistit, jestli je přijatá pseudonáhodná posloupnost nějak poškozena přenosovým kanálem, musíme originální PSP zachycenou pomocí rádiové části přijímače porovnat s přijímačem vygenerovanou pseudonáhodnou posloupností. Funkce vytvořená pro tento proces se jmenuje *frameCompare()*.

```
uint16_t frameCompare(uint8_t *frame)
{
    ...
    if(frame[0] != 127)
        return 64*8;
```

```

for( i = 1; i < frame[0] - 2 + 1/*-CRC+length*/; i++)
{
    tmp = frame[i]^PSPData[i];
    if(tmp != 0)
        errorCounter += bitCalcBuf[tmp&0x0f] + bitCalcBuf[tmp>>4];
}
//CRC
tmp = frame[i] ^ 0x6F;
if( tmp )
{
    errorCounter += bitCalcBuf[tmp&0x0f] + bitCalcBuf[tmp>>4];
}
tmp = frame[i + 1] ^ 0x37;
if( tmp )
{
    errorCounter += bitCalcBuf[tmp&0x0f] + bitCalcBuf[tmp>>4];
}
return errorCounter;
}

```

Funkce nám vrátí hodnotu chybných bitů, kde sumarizací této hodnoty získáváme vypisovanou hodnotu BEC (Bite Error Counter). Při každé zjištěné bitové chybě se nám zvýší FEC (Frame Error Counter) o jedničku. Bude-li BEC = 30, FEC se nám zvedne např. z FEC = 10 na FEC = 11 a BEC se nám zvedne o hodnotu 30 proti původní hodnotě.

S každým přijatým rámcem se nám FC (Frame Counter) zvýší o jedničku, aby zaznamenal přijetí dalšího rámce. Pomocí této vlastnosti a současného nastavení vysílače do režimu 10000 vyslaných paketů, můžeme velice krátkým měřením, přibližně 100 sekund, zajistit přesně stanovené podmínky z hlediska počtu vyslaných rámců a výsledek měření zaznamenat do tabulek a srovnat jednotlivé údaje naměřené z jiných míst.

```

void MLME_RX_ENABLEconfirm(void)
{
    params.start.panId = 0xDADA;
    channel = APP_START_CHANNEL - PHY_MIN_CHANNEL;
    while (!((1UL << (channel + PHY_MIN_CHANNEL)) & CHANNEL_MASK))
        channel = (channel + 1) & 0x0F;
    params.start.channel = channel + PHY_MIN_CHANNEL;
    params.start.panCoordinator = TRUE;
    params.start.coordRealignement = FALSE;
    MLME_STARTrequest(&params.start);
}

```

Další z funkcí potvrzuje přijetí rámce vysláním nastavených parametrů, které jsou totožné s vysílacími parametry. Jako hlavní identifikátor je určen panId, který nám dovoluje rozlišit jednotlivé sítě, a tak se vyvarovat zpracování nepotřebných dat z jiné sítě.

Protiklad k funkci odeslání dat, je funkce přijetí dat. V programu Receiver se o přijatá data postará funkce *sensTestAuxiliary_frameReceiveDone()*, kde **data* je ukazatel na přijatá data a umožňuje nám s přijatými daty pracovat.

```

void sensTestAuxiliary_frameReceiveDone(uint8_t *data, uint8_t nextLQI, uint8_t nextRSSI)
{
    ...
    tmpLQI = nextLQI;
    avaragedLQI = avaragedLQI - ((avaragedLQI - (tmpLQI << 7)) >> 5);
    outLQI = avaragedLQI >> 7;
    ...
    tmpRSSI = nextRSSI;
    avaragedRSSI = avaragedRSSI - ((avaragedRSSI - (tmpRSSI << 7)) >> 5);
    outRSSI = avaragedRSSI >> 4;
    ...

    frameCounter++;
    tmpErrors = frameCompare(data);
    if(tmpErrors)
    {
        frameErrorCounter++; bitErrorCounter += tmpErrors; }
}

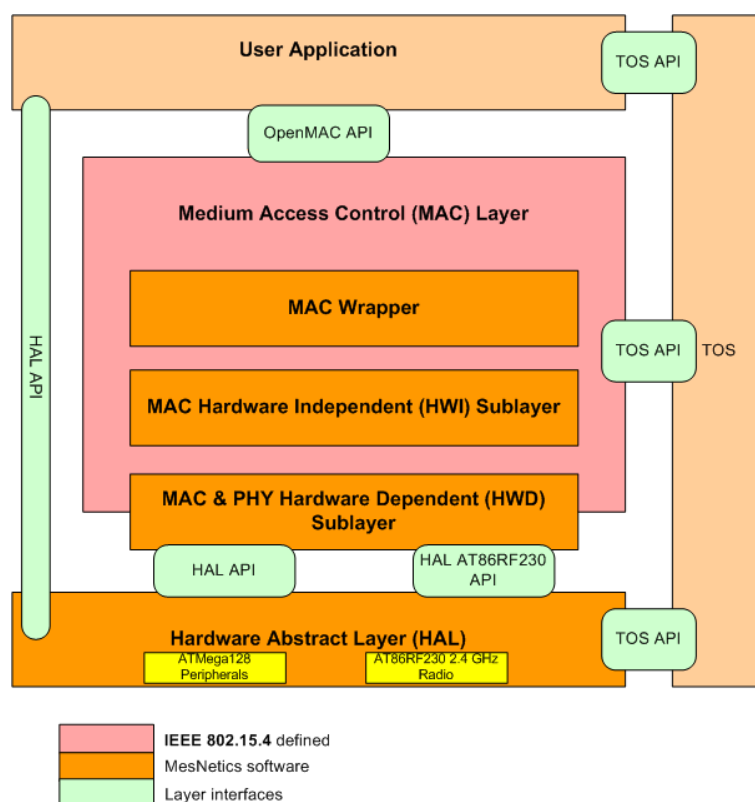
```

6.3 OpenMAC

Program Transmitter a Receiver jsou vzorové aplikace nabízené firmou Meschnetics. Jedná se o balíček OpenMac, který si lze zdarma stáhnout na stránkách MeschNetics [11].

OpenMac je od Meshnetics open source implementace IEEE802.25.4 Media Access Control (MAC) layer. MAC layer tvoří základ MeshNetics eZeeNet software stacku a poskytuje tak podporu peer-to-peer a síťové topologie typu hvězda.

MashNetics OpenMac chce umožnit uživatelům, kteří nevyžadují plnou funkčnost MeshNetics eZeeNet software stacku, vyvinout WSN aplikace na ZigBit moduly. Dále chce dovolit zkušenějším uživatelům upravovat OpenMAC ke konkrétnímu použití aplikací, poskytnout okamžitý start nad MAC Layer, který je v open source dostatečně okomentovaný. Snaží se poskytnout vhodný C API vývojářům neobeznámeným s TinyOS nebo nesC programovacím jazykem (technologii v jádru OpenMAC). Nabízí obraz směřovaný na podobné hardwarové platformy. OpenMAC je výrobní kód a je obsažený v prakticky nezměněné formě v MeshNetics eZeeNet softwaru. Graficky zobrazenou architekturu vidíme na Obr. 7.1[11].



Obr. 7.1: Architektura OpenMAC Layer

7 Řídící program


Obslužný software zajišťuje zpracování přijatých dat z modulů (XBee, ZigBit) a vypsání hodnot na připojené LCD. Program je nahráný v procesoru MC9S08GB60 a stará se o zobrazování a komunikaci s přijímacími moduly. Zvolený procesor MC9S08GB60 poskytuje dvě sériové linky a možnost připojit dva moduly k procesoru. Modul ZigBit je připojen na TXD2,RXD2 a modul XBee k RXD1. Pro možnost širšího využití přijímací zařízení je vybaven konektorem RS232 k propojení s PC. MC9S08GB60 obsahuje pouze dvě sériové linky, proto je nutné linku RxD1 přepínat pomocí jumperu RX1 mezi RS232 a XBee. Chceme-li využít ovládání procesoru MC9S08GB60 přímo z PC, použijeme zapojení s připojeným RS232 .

Hlavní program zastupuje *main.c* , který zkompletuje všechny potřebné header files pro správný běh programu, dále obsahuje globální proměnné využívané v programu, deklaraci přijímacího zařízení a hlavně funkci *main* ve které se provedou všechny potřebné kroky pro nastavení procesoru. Vývojový diagram programu nahráný v mikroprocesoru MC9S08GB60 umístěn v Příloha B.

7.1 Inicializace MCU

Postupně dochází k inicializaci procesoru, kde začínám výběrem zdroje hodinového signálu na interní oscilátor a správné nastavení je důležité pro nastavení SCI rozhraní na požadovanou přenosovou rychlost. Nastavením ICGC1 (viz Obr. 7.1) na hodnotu 0x44 zajistíme připojení interních hodin, frekvenční rozsah je zvolen pro vysoké frekvence (prescale P = 1) [1]


	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	RANGE	REFS	CLKS	OSCSTEN	0 ⁽¹⁾	0	
Write:								
Reset:	0	1	0	0	0	1	0	0

 = Unimplemented or Reserved

Obr. 7.1 ICG Control Register 1 (ICGC1)

Pomocí kontrolního registru ICGS1 (viz Obr. 7.2) a 1. bitem CLKST (Clock Mode Status) zkontrolujeme nastavení na Self-clocked a CLKST = 0x00[1].

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CLKST	REFST	LOLS	LOCK	LOCS	ERCS	ICGIF	
Write:								1
Reset:	0	0	0	0	0	0	0	0

 = Unimplemented or Reserved

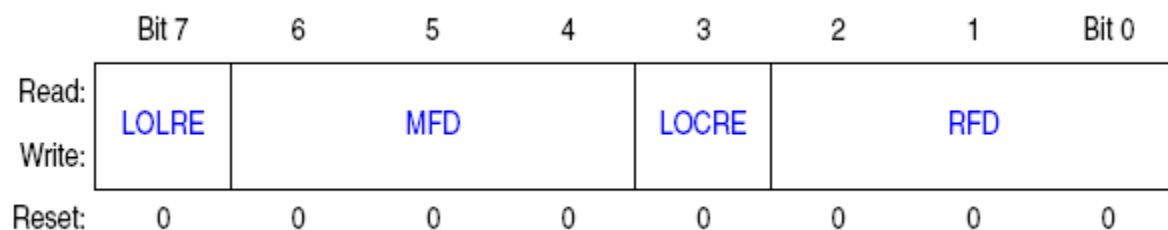
Obr. 7.2: ICG Status Register 1 (ICGS1)

Nastavením kontrolního registru [1] (viz Obr. 7.3) ICGC2_LOLRE = 1 a požadavek na restart procesoru při ztrátě hodinového signálu, MFD = 0x00 a násobení hodinového signálu 4x a RFD =

0x00 ě redukovaný dělič kmitočtu 1, tím zajistíme frekvenci sběrnice viz vztah (1), při nastavení CLKS = 0x01(viz br 7.1) ě FLL engaged, internal reference, získáváme hodnoty dle (1).

$$f_{BUS} = 2MHz * 4 = 8MHz \quad (1)$$

Nastavením CLKS = 0x00 nenastává žádná reakce na nastavení MFD a frekvence sběrnice bude nastavena $f_{BUS} = 4 MHz$.



Obr. 7.3: ICG Control Register 2 (ICGC2)

Takto nastavené hodnoty vložím do funkce *use_external_clock()*

```
void use_internal_clock()
{
    ICGC1 = 0x44;
    ICGC2_LOLRE = 1;
    ICGC2_MFD = 0x00; // násobička kmitočtu 4x
    ICGC2_RFD = 0x00; // redukovaný dělič kmitočtu 1
}
```

7.1.1 Inicializace SCI

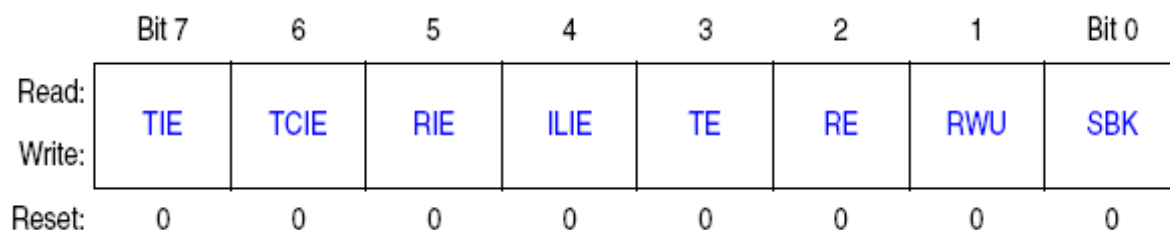
Komunikace mezi procesorem, modulem XBee, ZigBit a PC probíhá pomocí sériové komunikační linky SCI. Moduly obsahují program na měření kvality signálu. Výsledná naměřená data posílají po sériové lince do mikroprocesoru MC9S08GB60. Podle toho, jaký je zvolený modul, je inicializována a spuštěna sériová linka.

Abychom mohli začít využívat sériovou linku, je zapotřebí toto rozhraní inicializovat. To znamená nastavit všechny potřebné parametry, které nám zajistí správné přečtení dat z modulu XBee a ZigBit. Moduly vysílají data po sériové lince rychlostí 38400 bps, 8-bit data, bez parity a jeden stop bit (zkráceně 38400, 8N1 mode).

V předchozí části jsme si nastavili hodiny a zajistili $f_{BUS} = 4 MHz$. Za pomocí vztahu (2) vypočítáme hodnotu BR pro potřebnou rychlost 38 400 bps.

$$SCI_baud_rate = \frac{f_{BUS}}{16BR} \Rightarrow BR = \frac{f_{BUS}}{16SCI_baud_rate} = \frac{4 \cdot 10^6}{16 \cdot 38400} = 6,51_{10} \rightarrow 07_{16} \quad (2)$$

Nyní známe hodnotu BR, kterou nastavíme do registru SCIxBDL. Abychom mohli SCIx používat je zapotřebí spustit vysílání (příjem) pomocí registru SCIxC2. Pro snadnější hlídání příchodu dat, aktivujeme požadavek na přerušení při příjmu Rx na SCI1a SCI2 viz Obr. 7.4.



Obr. 7.4: SCI x Control Register 2 (SCIxC2)

Proto nastavením hodnoty `SCI1C2 = 0b00101100` zapneme Rx, Tx, RIE viz Obr. 7.4.

```
void SCIIInit(void) {           //povolení a nastavení SCI1
SCIxBDH = 0x00;
SCIxBDL = baud38400; // hodnota nastavující SCI na 38400 tedy BR = 0B;
                        // pro FBUS = 8MHz
SCIxC2 = initSCIxC2;          // spustení SCI - Tx,Rx, RIE
}
```

Pro signalizaci činnosti (stavu) je zapotřebí inicializovat *Portu D* na procesoru jako výstupní a připojení LED.

```
void LEDinit(void) { //inicializace LED
    LEDxDIR = 1;      // nastavení pinu na výstup pro LED
    LEDx = 0;         // nastavení hodnoty pinu 0 = rozsvícená led
}
```

Ovládání procesoru je zajištěné pomocí čtyř tlačítek a připojených k Portu A se zapojenými pull-up rezistory označených jako 1:>, 2:,<, 3:mode, 4:set. Označení pomocí čísel zajišťuje přehlednou navigaci na LCD a při popisování v manuálu srozumitelnost.

```
void PBinit(void){ //vstupy
    PBxDIR = 0;      // nastavení pinu na vstup
//pull up on
    PBxPU = 1;       // připojení PU rezistoru
//Interrupt enable pro tlačítka
    PBxIE = 1;       // povolení prerušení od tlačítek
    KBIISC_KBIE = 1; //povolení KBI prerušení
}
```

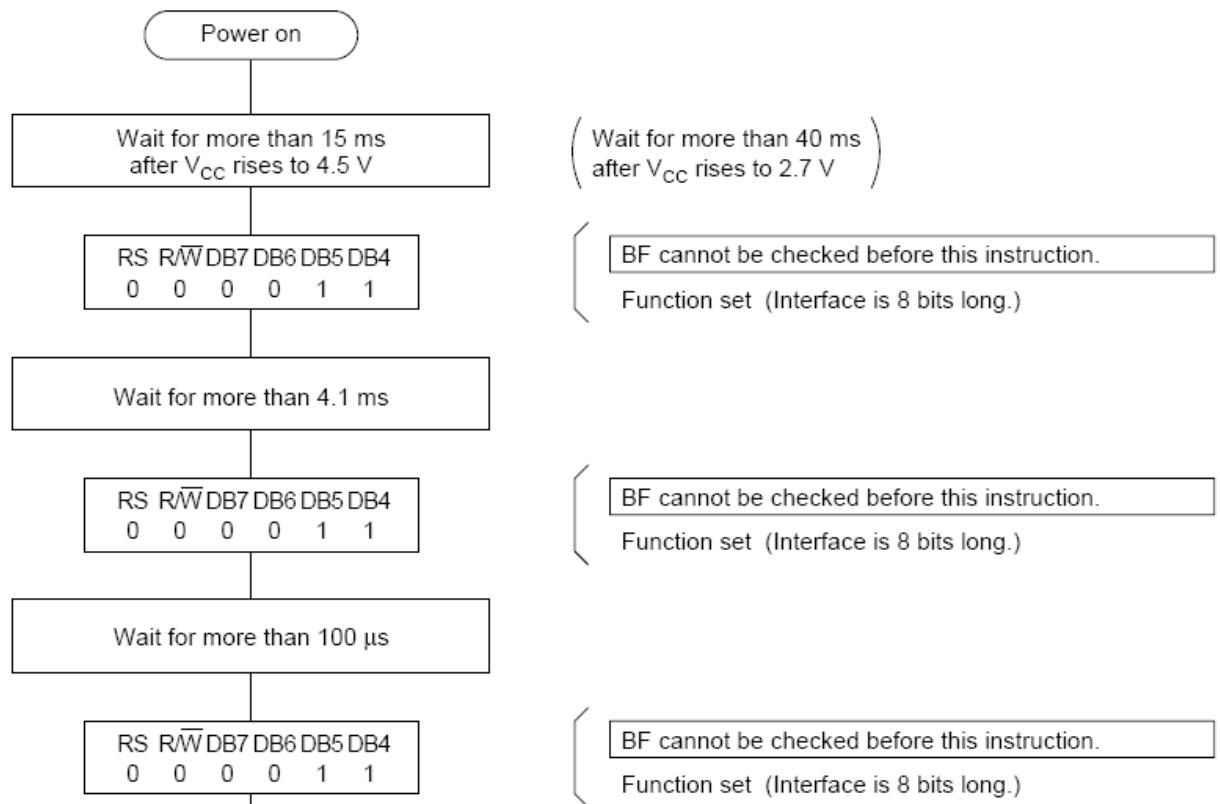
Nyní máme připravený procesor pro komunikaci a můžeme přijímat data a příkazy pro ovládání přijímacího zařízení.

7.2 Ovládání HD44780 (LCD1602)

Zobrazení získaných dat zajistíme připojeným LCD zobrazovacím zařízením o dvou řádcích a šestnácti znacích, který obsahuje řadič HD44780. Komunikace mezi procesorem MC9S08GB60 a řadičem HD44780 je zprostředkována pomocí 4-bitové datové sběrnice. Abychom mohli začít komunikovat s LCD zařízením, musíme provést inicializaci pro 4-bitovou komunikaci, kterou nám zajistí funkce `LCD_init()`.

7.2.1 Inicializace LCD

Prvním důležitým krokem je vyslání horní poloviny slova na dvakrát, aby řadič HD44780 zaznamenal pokus o 4-bitovou komunikaci. Důležité je dodržet 4,1 ms zpoždění mezi polovinami viz Obr. 7.5, aby byl řadič HD44780 schopný reagovat na další zápis dat. Poté následuje vyslání celého 8-bitového slova ve dvou částech. Tímto jsme zajistili komunikaci mezi LCD a procesorem.



Obr. 7.5: Inicializace pro 4-bitovou komunikaci

```

void LCD_init(void) {
    delayms(d15ms); //15ms na start zpoždění

    RS_inda = instrukce;           // vysílání instrukci -> RS = 0
    DB_port = 0x02;                // pro 4bitovou komunikaci a vyslat horní polovinu 2x.
    E_enable = 1;                  // potvrzení dat na výstupu
    delay(d1us);
    E_enable = 0;

    delayms(d4_1ms);               // nutno cekat požadovanou dobu      4,1ms

    /* Vyslání kompletní inicializace po 4-bit lince */
    //////////////////////////////////////
    pisLCD(0x28); // potvrzení 4-bitové inicializace(kompletní), nastavení
                // na dva řádky, s fontem font 5x8
    delay(d100us); // tedy ma byt 100us

    //////////////////////////////////////
    pisLCD(0x06); // posun kurzoru vpravo, posun textu ne

    //////////////////////////////////////
    pisLCD(0x0C); // zapne displej. vypne kurzor, vypne blikání

    ...
  
```

Nyní pokračujeme v inicializaci nastavení dvouřádkového zobrazování s fontem písma 5x8 \Rightarrow 0x28_{hex} a následnou prodlevou 100 μ s doporučenou z technické dokumentace viz [2] nebo Obr. 7.5. Pro správné vypisování textu je důležité zadat posun kurzoru vpravo a posun textu vypnout \Rightarrow 0x06_{hex}. Nyní jako poslední variantu upravíme zobrazování kurzoru vypnuto, blikání vypnuto a zapnout displej \Rightarrow 0x0C_{hex}. Potřebujeme-li upravit některá z těchto nastavení, přesný formát instrukcí potřebných pro ovládání HD44780 najdeme v Příloze A.

7.2.2 Vypisování textu

Máme-li inicializovaný displej, můžeme na LCD posílat data, která potřebujeme zobrazovat. Zápis dat na LCD probíhá následovně. Jeden z řídících vodičů nám určuje, zda-li vysíláme data nebo instrukce. V našem programu je označený jako *RS_inda*. Nastavíme-li hodnotu *RS_inda = data*, řekneme tím řadiči, že budeme vysílat data a že bude vybírat z tabulky znaků viz Tab. 5.2.

Výslednou funkci pro zapisování jednotlivých znaků je *LCD_char()*. Pro tuto funkci je nastaveno vysílání dat a poté zavoláme funkci *pisLCD()*, která nám odešle naše data pro 4-bitovou komunikaci.

```
void LCD_char(char znak){
    RS_inda = data;
    pisLCD( znak );
    RS_inda= instrukce;
}
```

Tímto jsme si vytvořili funkci pro výpis jednotlivých znaků. Jelikož budeme potřebovat zapisovat i celá slova na LCD, je zapotřebí vytvořit funkci pro výpis řetěze znaků, kterou nazveme *LCD_string()*.

```
void LCD_string(char *Textik){
    int i;
    int leng = strlen(Textik); // zjisti delky vkladaneho retezce
    RS_inda = data;           // indikace prenosu dat!

    for(i=0; i<leng; i++){
        pisLCD(Textik[i]);    // jednotlivy vipis znaku na LCD
    }
    RS_inda = instrukce;      // opetovne nastaveni pro zapis instrukci
}
```

Začneme tím, že funkce bude mít parametr, který jí potřebujeme předat. Tento parametr bude typu *char* a bude představovat celé pole znaků neurčité délky. Pro zjištění délky vkládaného pole využijeme funkci *strlen()*, která nám zjistí maximální počet znaků v poli a poslouží nám jako koncová hodnota pro cyklus *for*. Nyní nám stačí nastavit *RS_inda* pro vysílání dat, kde v cyklu *for* zahájíme jednotlivé odesílání znaků z pole *Textik[i]* a na konci funkce opět změníme *RS_inda* na instrukce.

7.2.3 Potvrzování dat pro LCD (pin Enable)

Výše uvedené funkce nám ukazují, jak postupovat, chceme-li odesílat data na LCD s řadičem HD44780. Zde si ukážeme, co je zapotřebí pro potvrzení dat, které chceme odeslat, protože bez potvrzovacího impulsu (Enable) bychom vysílali data na datovou sběrnici, aniž by došlo k jejich následnému potvrzení. Z tohoto důvodu je v každé funkci voláno *pisLCD()*, která nám zajistí nastavení požadované poloviny slova na datovou sběrnici a následné potvrzení pulsem o délce cca 1 μ s na pinu E (Enable).

```
void pisLCD(char pis){
    DB_port = pis >> 4; // nastaveni horní poloviny dat na port B
    LCD_E();           // potvrzení dat na sběrnici
    DB_port = pis;     // nastaveni dolní poloviny dat na port B
    LCD_E();           // potvrzení dat na sběrnici
    delay(400us);      // prodleva po vykonání zápisu
}
```

Vždy se jako první vysílá horní polovina slova, která se potvrdí a vzápětí je poslána dolní polovina slova. Po potvrzení dolní poloviny slova je zapotřebí zahrnout prodlevu delší jak 40 μ s, po kterých je řadič HD44780 zaneprázdněn zpracováváním dat.

7.2.4 Odřádkování

Využíváním druhého řádku je zapotřebí i přepínání polohy kurzoru na tento řádek. Podle řadiče HD44780 viz [2] je adresa druhého řádku 40_{hex} a ne 14_{hex} jak by bylo zapotřebí. Na tento problém narazíme, když budeme potřebovat zapsat delší text než je kapacita znaků jednoho řádku. Proto se podíváme, jak se dá tato adresa měnit. V příloze A jsou zobrazeny instrukce pro zápis do DDRAM (Displej Data RAM). V Tab. A vidíme, že pro změnu adresy v DDRAM je zapotřebí vyslat instrukci s aktivní DB7 + hodnota adresy. Nyní víme že nejnižší hodnota slova pro změnu adresy DDRAM je 80_{hex} , která odpovídá počátku displeje (levý horní roh). Nyní víme hodnotu instrukce pro počátek adresy displeje a víme i hodnotu adresy druhého řádku viz Tab. 7.1, následným sečtením získáme výslednou hodnotu adresy viz (3).

$$Hodnota_instrukce_2.\text{řádku} = 80_{hex} + 40_{hex} = C0_{hex} \quad (3)$$

Vyslání instrukce můžeme provést za pomoci funkce *pisLCD()*, kde funkci předáme hodnotu instrukce odpovídající druhému řádku $C0_{hex}$. Tímto způsobem si můžeme zjistit všechny pozice na displeji. Pro ujasnění hodnot adres na řádcích nám poslouží Tab. 7.1.

Tab. 7.1: Adresa jednotlivých pozic dvouřádkového displeje

Řádek	Adresa																
1.	0	1	2	3	4	5	6	7	8	9	10	1A	1B	1C	1D	1E	1F
2.	40	41	42	43	44	45	46	47	48	49	5A	5B	5C	5D	5E	5D	5F

7.3 Zpracování naměřených dat

MCU MC9S08GB60 vyhodnocuje zprávy vysílané po sériové lince z modulů MaxStream, MeshNetics. Zpráva obsahuje číslo odeslaného rámce, chybu rámce, bitovou chybu, kvalitu přenosové trasy a sílu signálu. Zpráva je odeslána jako řada znaků a pro vyhodnocování a zobrazování na LCD je zapotřebí zprávu rozdělit na jednotlivé části.

FC=87417 FEC=179 BEC=5695 LQI=254 RSSI=-65dBm(08)

Postačí nám soustředit se na jednotlivá čísla a ty separovat od zbytku řetězce. Víme také, že pozice písmen v řetězci zůstává stejná, tedy pozice mezi znakem '=' a následujícím písmenem má vždy stejný počet míst mezi sebou. Nyní se zaměříme na hodnotu FC (Frame Counter). Pro zjednodušení si v řetězci označíme jednotlivé pozice čísly.

1
2
3
4
5
6
7
01234567890123456789012345678901234567890123456789012345678901234
FC=87417 FEC=179 BEC=5695 LQI=254 RSSI=-65dBm(08)

Nyní přesně známe pozici každého znaku. Indexy pro počáteční a koncovou polohu tedy odečteme \Rightarrow start = 3 a konec = 14 a známe maximální délku čísla, které bude určovat opakování cyklu *for*. Podíváme-li se na další hodnoty FEC, BEC, zjistíme, že disponují stejným počtem pozic pro zobrazení čísla, a proto využijeme stejný cyklus se stejnou délkou jako v našem prvním případě FC.

```
void FC_rozdel(void){
int i;
int chSCI = 3; //start na hodnotě 3, kde začíná číslo

for(i=0; i<12; i++){ // délka načítaného řetězce
```

```

        chSCI += i;
        konst_Rx_ch[i] = (bufferSCI_Rx[chSCI]);
    }
}

```

Důležitá je proměnná *chSCI*, která nám ukazuje na první pozici znaku, a cyklus, který už automaticky pokračuje od pozice *chSCI*, dokud nenačte celý řetězec požadované délky. Díky této proměnné můžeme využít tento cyklus beze změn i pro načtení hodnoty FEC a BEC. Pro načítání hodnoty LQI a RSSI je postup stejný, jenom se upraví délka řetězce a pozice počátečního znaku řetězce.

Získáme-li takto řetězec znaků, je zapotřebí ho převést na číselnou hodnotu, protože získaný řetězec znaků je typu *char* a ne hodnota typu *int*. Můžeme zde vyzkoušet převod řetězce znaku funkcí *atoi()*, ale během mého testování v reálně běžící aplikaci tento proces vždycky havaroval a zastavil chod programu. Proto jsem zkoumal, jak daný řetězec znaků převést na typ *int*.

Věděl jsem, že každé písmenko, znak a číslice mají přiřazenou ASCII hodnotu. Podíval jsem se tedy do ASCII tabulky a našel hodnoty čísel 0 – 9. Byly řazeny za sebou a vzestupně, což usnadnilo převod čísel. Nula jako číslice v ASCII tabulce má hodnotu 48_{dec} a číslice devět má hodnotu 57_{dec}. Nyní když známe hodnotu nuly a můžeme ji tak odečíst od aktuální hodnoty dané číslice. Získáme tak číslo rovnající se hodnotě zobrazeného znaku, kterou jsme vnímali jako číslo.

```

int str2int(char *dd){
    int i,j,i_start, cislo, vysledek =0;
    int delkka, delkka_bezmezr;
    delkka = strlen(dd);
    delkka_bezmezr = delkka; // promenna pro delku retezce
                                // bez mezer

    for(j=0; j < delkka; j++){ // cyklus pro zjisti poctu mezer
                                // v celem retezci
        if( dd[delkka - j] == ' ') delkka_bezmezr--;
    }

    if(delkka_bezmezr > 4){ // je-li delka prevadeného retezce
        i_start = delkka_bezmezr - 4; // delsi nez 4mistne cislo, dojde
    } else i_start = 0; // k nastaveni retezce na start posledni 4 mista

    for(i=i_start; i<delkka_bezmezr; i++){ //cyklus pro ukazani na potrebné místo
        if(dd[i] >= '0' & dd[i] <= '9' ){ // podminka k vymezení rozsahu pro
            vysledek *= 10; // zpracovani pouze cislic
            cislo = dd[i] - '0'; // odedeni ASCII hod. od znaku a dostaneme cislo
            vysledek += cislo;
        } //if
    }
    return vysledek; // funkce vrati hodnotu vysledek
}

```

Bohužel na vyslané zprávě jsou doplněny mezery mezi koncem čísla a prvním znakem za číslem, proto je zapotřebí tyto znaky vypustit. Vytvoříme si tedy cyklus pro odstraňování mezer. V cyklu nastavíme zjištěnou délku řetězce s mezerami pomocí *strlen()*, projdeme celý řetězec, abychom zjistili počet mezer a mohli je odstranit. Při každé nalezené mezeře dekrementujeme proměnnou *delkka_bezmezer*. Tímto jsme si zjistili délku čísla, které máme uložené v řetězci a přistoupíme na další krok, kterým převedeme znak na číslo.

Vymezíme si délku řetězce, kterou budeme převádět. Tedy koncová pozice *delkka_bezmezer* a výchozí pozice o čtyři řády nižší. Tato funkce nám bude sloužit na převod řetězce na číslo s maximální hodnotou 9999. Teď přistoupíme ke čtení aktuálního znaku z řetězce. Načteme hodnotu a otestujeme ji, zda-li se jedná skutečně o číslici a od znaku odečteme ASCII hodnotu nuly(48_{dec}). Výsledné číslo je poté přičteno do proměnné *vysledek*. V dalším kroku je průběh zcela totožný, ale *vysledek* je hned na začátku násoben deseti, aby došlo k posunu o jeden řád a přičtení nového čísla k *vysledek*. Tímto

způsobem získáme naše požadované číslo uloženo jako datový typ *int* a možnost s ním pracovat jako s číslem. Funkce s názvem *int2str()* nám tedy uskuteční převod znaku na číslo.

7.3.1 Indikace ztráty signálu

Přijímá-li přijímač signál, zobrazuje aktuální informace na displej. Dojde-li ke ztrátě signálu, bude na LCD blikat nápis *!Signal ztracen!*. Ztrátu signálu detekujeme podle hodnoty FC, začne-li se hodnota FC opakovat pořád dokola, nastala ztráta signálu. V předchozí části jsme rozdělili řetězec znaků a převedli na číslo.

```
void LostSig(void){
    FC_rozdel();          // rození retezce ziskaneho z SCI2
    FC = str2int(konst_Rx_ch); // prevod ziskane hodnoty ve formatu char na int
    if(fc_temp == FC){    // bude-li prechozi hodnota stejná jako poslední
        LED4 = ~LED4;    // doslo ke ztrate signalu a blikani LED4
        LCD_2radek();
        if(LED4 == 0) LCD_string("!Signal ztracen!"); // vypis na druhy radek
        else LCD_string(" "); // premazani LostSig
    } else LED4 = 1; //vystraha ztraty signalu // musi byt jinak zhasnuta
    fc_temp = FC;        // ulozeni nove hodnoty do predchozi prom.
}
```

Dále stačí porovnávat proměnné a při shodě vypsat následující varovné hlášení. Displej sice indikuje ztrátu signálu, ale zobrazený nápis není tolik viditelný, ikdyž jeho výpis je přerušovaný. Proto je indikace ztráty signálu podpořena vizuálně pomocí LED umístěné na přijímací zařízení.

7.3.2 Volba přijímacího modulu

Přijímací zařízení je opatřeno dvěma patičkami pro dva různé moduly. Je možné připojit bezdrátový modul MaxStreamu (XBee) anebo MeshNetics (ZigBit). Nelze tyto moduly zaměnit, protože zapojení konektoru není zcela stejné.

Zapneme-li přijímací zařízení, vypíše se úvodní obrazovka s názvem přijímacího zařízení a jaká verze softwaru je nahráná v řídicím procesoru MC9S08GB60. Po zobrazení úvodní obrazovky následuje dotaz na výběr modulu, který budeme chtít používat.

```
void Module(void){
    DisableSCI2();
    SCITransmitStr("\n\nVyber zarizeni\n\n 1:ZigBit nebo 2:XBee\n");
    SCITransmitStr("klavesa h - help\n\nvolba:");
    kb_stop = 1; // povoleni nasledujiciho cyklu
    while(kb_stop == 1) { // cekani na vzber pomoci tlacitek 1(<) a 2(>)

        /* PB0 ZigBit */
        if(PB0 == 0 | SCIData[0] == '1') { // tlacitko 1 vyber ZigBit
            modul = ZigBit;
            SCITransmitStr("\nVybrany prijmac: ");
            SCITransmitStr("ZigBit\n\n");
            kb_stop = 0; // promenna ukonceni cyklu
        };

        /* PB1 XBee */
        if(PB1 == 0 | SCIData[0] == '2') { // tlacitko 2 vyber XBee
            modul = XBee;
            SCITransmitStr("\nVybrany prijmac: ");
            SCITransmitStr("XBee\n\n");
            kb_stop = 0; // promenna ukonceni cyklu
        };

    }
    if(SCID == '1') menu_status++; // pro ovladani menu i pres SCI
    if(SCID == '2') menu_status--;
    if(SCID == 'c' | SCID == '1' | SCID == '2') EnableSCI2(); //povoleni preruseni, jen v
    poripade jedna-li se o c,1,2
}
```

```

menu_status = Hlavni_menu1;
}


```

Výběr modulu je prováděn pomocí místních tlačítek na přijímacím zařízení. Budeme-li chtít používat modul XBee, bude zapotřebí přepnout jumper RX1 na XBee, abychom připojili sériovou linku z procesoru MC9S08GB60 k XBee. Tímto způsobem bude sériová linka k počítači připojena stále, ale jenom z vysílací strany přijímacího zařízení, tedy jeden směr komunikace přijímacího zařízení k PC. Řídící program obsahuje i drobnou nápovědu a možnost ovládání přijímacího zařízení přes PC, ale v tomto případě ovládání nebude fungovat.

7.3.3 Přerušení tlačítek

Ovládání přijímacího zařízení je možné pomocí tlačítek umístěných na desce plošného spoje. Přijímací zařízení obsahuje celkem čtyři tlačítka umístěná na přední levé straně přijímacího zařízení. Pomocí tlačítek se budeme pohybovat v menu a na displeji můžeme pozorovat aktuální vybranou položku. Víme, že používáme procesor MC9S08GB60, který nám nabízí funkci *Keyboard interrupt*. Využitím této funkce pro přerušení nebudeme muset skenovat port s připojenými tlačítky, kde můžeme procesor při stavu nečinnosti zastavit a šetřit baterii. Aktivování funkce Keyboard interrupt podle [1] najdeme v registru KBIISC viz Obr. 7.6, kde pro samotnou aktivaci a přerušení z klávesnice je 1.bit KBIE. Nastavením KBIE na hodnotu 1 a KBIMOD na 0 zajistíme aktivaci přerušení pro detekci hrany. Funkce KBI je možné využívat pouze tehdy, budeme-li mít tlačítka připojená na port A.

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	KBEDG7	KBEDG6	KBEDG5	KBEDG4	KBF	0	KBIE	KBIMOD
Write:						KBACK		
Reset:	0	0	0	0	0	0	0	0

 = Unimplemented or Reserved

Obr. 7.6: KBI Status and Control Register (KBIISC)

Víme, že budeme používat pouze čtyři tlačítka připojená na PTA2..PTA5. Podle registru KBIISC viz Obr. 7.6 je možné na tlačítkách připojených k PTA4 a PTA5 nastavit za jakých okolností chceme vygenerovat přerušení. Kdybychom nastavili KBEDG4..5 (*Keyboard Edge Select*) na log. 1 zajistili bychom generování přerušení na náběžnou hranu a ustálení na vysoké úrovni. Podle schématu zapojení tlačítek k procesoru, potřebujeme detekovat sestupnou hranu a nízkou úroveň, a proto ponecháme nastavení bez změn.

Využíváme-li obsluhu klávesnice, je zapotřebí povolit (aktivovat) potřebné piny portu A, které budeme potřebovat pro generování přerušení. Registr pro povolení potřebných pinů se nazývá KBI1PE (KBI Pin Enable Register) viz Obr. 6.7 a zapsáním jedničky do potřebných kolonek bitů registru aktivujeme příslušný vstupní pin portu A. V našem případě vynecháváme první dva piny a hodnota pro aktivování pinu 2..5 bude 0x3C_{HEX}(0b00111100_{BIN}).

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	KBIPE7	KBIPE6	KBIPE5	KBIPE4	KBIPE3	KBIPE2	KBIPE1	KBIPE0
Write:								
Reset:	0	0	0	0	0	0	0	0

Obr. 7.7: KBI Pin Enable Register (KBI1PE)

Zmáčkneme-li tlačítko, vygeneruje KBI přerušení a nastaví se hodnota KBF = 1, kterou je zapotřebí po provedení přerušení vynulovat viz Obr. 7.6. Nulování KBF se provádí pomocí zápisu jedničky do KBACK. Takto nastavený a obsluhovaný kontrolní registr KBIISC nám zajistí využití přerušení z klávesnice.

```
interrupt 22 void Tlacitko(void){
    DisableInterrupts;           //vypnutí prerušení
    if(PB3 == 0) LCD_init();     //zatím případná inicializace
    if(PB2 == 0) {               //změna přijímacího modulu
        if(modul == ZigBit) modul = XBee;
        if(modul == XBee) modul = ZigBit;
        menu_status = Hlavni_menu0; //nastavení pro nový výpis prvního řádku
    }

    if(PB0 == 0) {
        menu_status++;
    }

    if(PB1 == 0){
        menu_status--;
    }

    if(menu_status < 1) menu_status = 6; //ovládání zobrazovacího menu
    if(menu_status > 6) menu_status = 0; //v rozmezí 1 - 6

    KBIISC_KBACK = 1;           //potvrzení vykonaného přerušení tlačítek
    EnableInterrupts;           //povolení všech přerušení
    kb_stop = 1;               //povolení vykonání smyčky v module()
    }                           //kde se čeká na stisk klávesnice
```

Aktivujeme-li si přerušení, je zapotřebí tuto rutinu zpracovat a říct procesoru, co bude vykonávat. Všechny přerušení pro procesor MC9S08GB60 mají přidělený vektor přerušení, který určuje prioritu přerušení např. restart procesoru je na nultém místě a RTIE (Real-time interrupt) je na posledním tedy 25. místě. Pro práci s klávesnicí nás zajímá vektor přerušení 22 (Vkeyboard1 KBI KBF KBIE Keyboard pins) viz příloha B.

Známe-li vektor přerušení nebo jeho jméno, můžeme vytvořit funkci, která se bude volat při generování přerušení z klávesnice. Zápis funkce přerušení klávesnice provedeme obdobně, jako bychom psali normální funkci, jenom jako první vložíme slovo interrupt a vektor přerušení 22 *interrupt 22 void Tlacitka(void){}*. Nyní můžeme přistoupit k testování tlačítek, abychom zjistili zmáčknuté tlačítko. Menu obsahuje konkrétní počet položek, a proto zde budeme hlídat přetečení nebo podtečení, abychom zajistily rotaci v menu. Přímo na začátku funkce je deaktivace přerušení, aby náhodou nedošlo k přerušení automatickým přijímáním dat z modulu. Po provedení obsluhy klávesnice dojde k opětovnému zapnutí přerušení a nulování KBF příkazem *KBIISC_KBACK = 1*, které takto deaktivujeme a povolíme další obsluhu klávesnice.

7.3.4 Menu

Z důvodu malé kapacity zobrazovací jednotky je zapotřebí vytvořit menu, které nám zajistí srozumitelný a jednoduše ovladatelný výpis dat na zobrazovací jednotku. Prvním krokem si rozdělíme přijímanou zprávu na pět částí *ě* hodnota = položka menu. Přijímací zařízení má dva moduly mezi kterými je zapotřebí při spuštění vybírat *ě* tedy další část menu. Nyní máme menu rozdělené na šest částí a můžeme začít tvořit funkce *menu()*. Jako vhodnou variantu vytvoříme menu na principu stavového automatu, kde se bude pomocí indexů vybírat jednotlivá položka menu. Za pomoci *case* si vytvoříme kostru, kterou můžeme kdykoliv upravit (přidat položku, odebrat položku) a vytvořit tak finální podobu menu.

```
void menu(void){
    switch(menu_status){

        case Hlavni_menu1:           //výpis prvního řádku na LCD
            piLCD( 0x80 );
```

```

        if(modul == ZigBit) {
            LCD_string("Prijimac: ZigBit");
        }
        if(modul == XBee) {
            LCD_string("Prijimac: XBee ");
        }
        menu_status = FC_menu;

        break;

    case FC_menu:                //FC vypis 2.radek
        LCD_2radek();           //druhy radek
        LCD_string("FC: ");
        FC_rozdel();
        LCD_string(konst_Rx_ch);

        break;

    ...
    ...
    ...

    case RSSI_menu:              //RSSI vypis 2.radek
        LCD_string("RSSI: ");
        RSSI_rozdel();
        LCD_string(konst_Rx_ch);
        LCD_string("dBm ");
        break;
    }
}

```

Hlavní menu vypisuje z jakého modulu se data získávají(ZigBit, XBee). Další kategorie FC_menu, FEC_menu, BEC_menu, LQI_menu, RSSI_menu jsou vypisovány na druhý řádek. Za pomoci předem vytvořených funkcí získáváme hodnoty, které potřebujeme a necháváme je vypisovat na displej. Hodnoty jsou měněny právě tak často, jak jsou odesílány z daného modulu (cca 1s).

7.3.5 Funkce main

Snažil jsem se, aby funkce *main* vypadala co nejjednodušeji. Funkce je rozdělena do několika částí: inicializace MCU, úvodní obrazovka, výběr modulu, nekonečná smyčka obsahující kontrolu plnosti SCIbufferu a ztráty signálu. Na konci tohoto cyklu dojde k zastavení procesoru, který vzbudí pouze výše uvedené přerušení.

```
void main(void) {

    mcu_init();
    use_external_clock(); /* switch clock sources */
    /***/
    /* Inicializace */
    /***/
    /* LED */ /* BUTTON */
    LEDInit();
    PBInit();

    /* SCI 1,2 */
    SCI1Init();
    SCI2Init();
    DisableSCI2();

    SCITransmitStr("\n\n *** Ridici MCU ***\n\nMCU.....Spusteno\n");

    /* LCD */
    LCD_init();

    /***/
    /* Uvodni obrazovka + SCI Text */
    /***/
    LCD_string("-Kvalita Prijmu-");
    LCD_string(" ZigBit x XBee");
    SCITransmitStr(verze);
    ...

    Module();
    EnableInterrupts; /* enable interrupts */

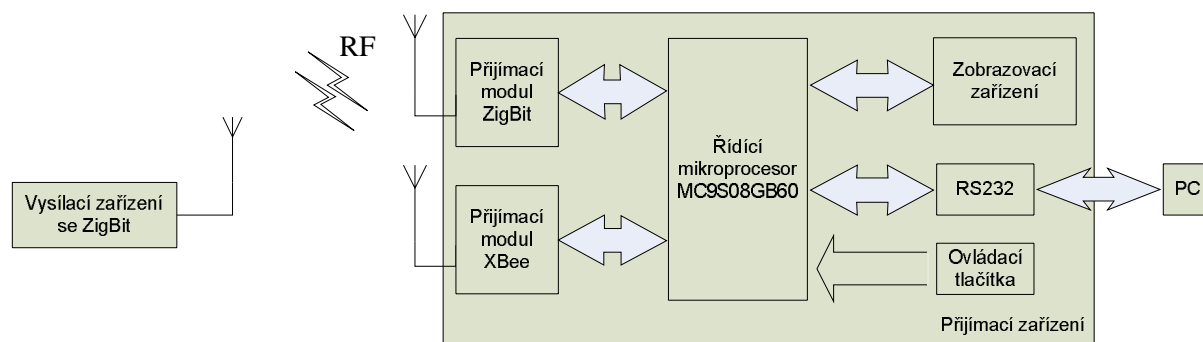
    /***/
    /***/ Sycka /***/
    /***/
    for(;;) {

        if(bufferFull == 1){ //je-li buffer plny dojde k povoleni
            //vypisu na LCD pripadne SCI
            menu(); //volani MENU pro vzpis na LCD
            bufferFull = 0; //nulovani priynakove promenne
            LostSig(); //test detekce ztraty signali
        }
        ...

        _asm wait //cekani na preruseni
    } /* loop forever */
}
```

8 Konstrukce

Pro využití zařízení v praxi je zapotřebí navrhnout desku plošného spoje (dále jen DPS), která nám zajistí vodivé spojení mezi konkrétními piny jednotlivých součástek, mechanickou pevnost a konkrétní rozmístění součástek využitých pro funkčnost zařízení. Výsledkem budou dvě osazené desky DPS. První s funkcí přijímače, vyhodnocující kvalitu signálu zobrazovanou na dvouřádkovém LCD s možností připojení k PC pomocí RS232 sériové linky. Druhá DPS slouží jako vysílací zařízení, vysílající sekvenci dat, které jsou zpracovávány v přijímacím zařízení a vyhodnoceny. Jako program pro navrhování DPS jsem si zvolil Eagle 4.16r2 lite s použitím do maximálních rozměrů DPS 100 x 80 mm, max. dvě signální vrstvy (Top a Bottom) [4] Verzi „lite“ je možné zdarma stáhnout z internetových stránek viz [5]. Výsledkem bude jedno vysílací zařízení a jedno přijímací zařízení osazené dvěma přijímacími moduly.



Obr. 8.1: Blokový diagram zařízení pro měření kvality signálu

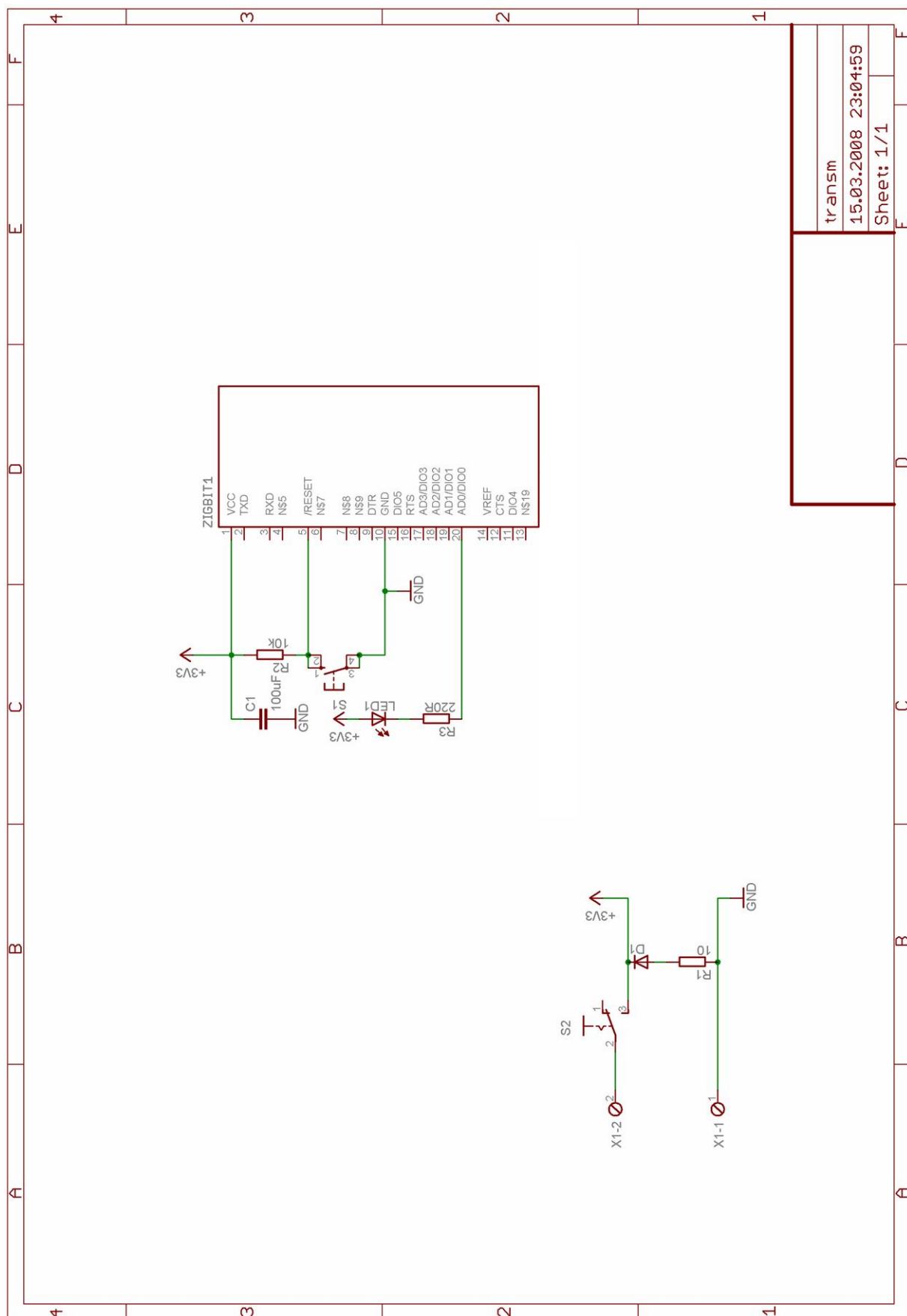
8.1 Vysílací zařízení

Pro využití v praxi je vhodné, vyrobit vysílací zařízení, jako menší a zajistit tím mobilitu výrobku a možnost snadného uchycení nebo umístění v požadovaném bodě vysílání. Náročnost výroby malého vysílacího zařízení není velká. Hlavní a aktivní vysílací prvek je výrobek firmy MeshNetics jako ZigBit module s Dual Chip Antenna. Jeho rozměry 24 x 13,5 x 2,8 mm zajišťují výrobu malého zařízení. Zdroj energie bude ze 2 x AAA článků uchycených v držáku na 2 AAA s připájenými vývody pro připojení k svorkám na DPS. Tímto máme dané rozměry DPS, která překrývá horní stranu držáku baterií.

Moduly ZigBit jsou umístěné na redukci z verze ZigBit SMD a XBee verzi rozmístění vývodů, kde šířka řad vývodů je cca o 2 mm užší než držák baterií. Další součástky můžeme umístit na druhou polovinu DPS, protože délka redukce je cca polovina držáku baterií, tak na tlačítko, vypínač a svorku pro připojení vodičů je vyhrazena druhá polovina.

8.1.1 Schéma vysílače

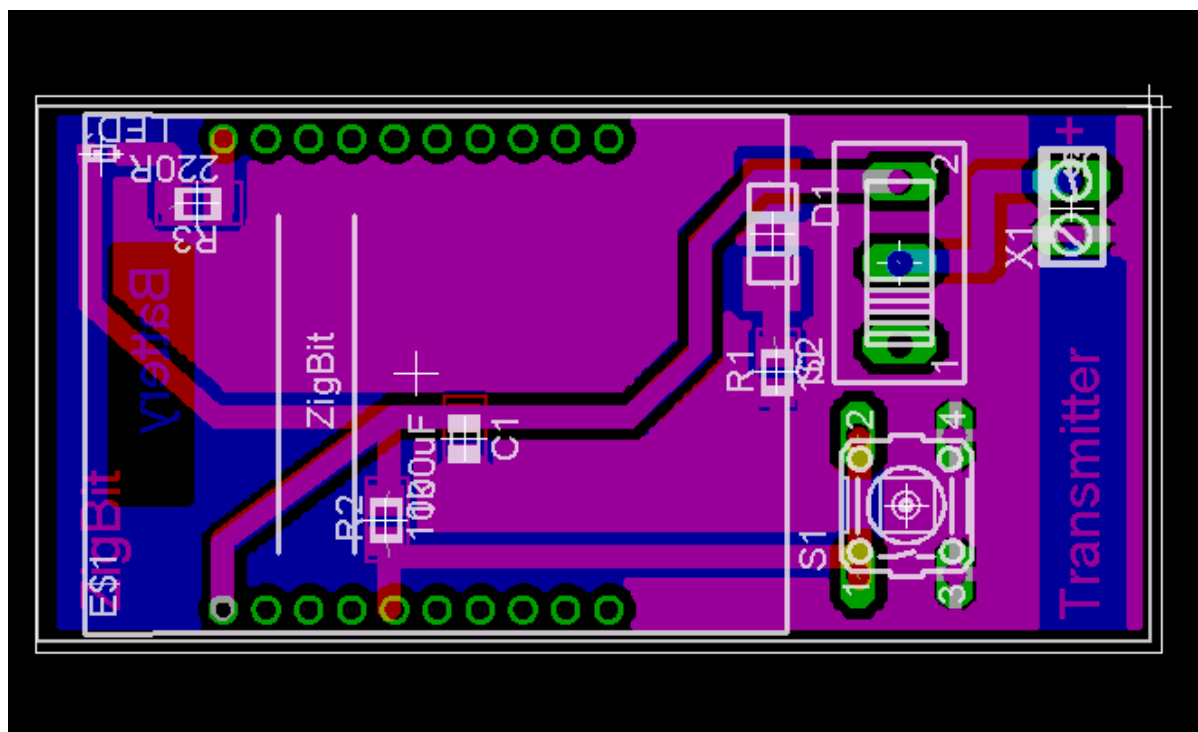
Prvním krokem pro nakreslení schématu je znalost součástek, které budeme používat. Nejdůležitější součástky pro chod zařízení jsou ZigBit od firmy MeshNetics, přepínač, svorka na připojení zdroje, tlačítko pro zajištění případného resetu (doplněnek), připojení RESET-u na VCC přes odpor a blokuji kondenzátor proti zemi, LED dioda pro indikaci stavu programu a odpory doplňující LED. Dále jsem v napájecí části umístil diodu (BYS10 [6]) v závěrném směru doplněnou o odpor malé hodnoty, které zde můžeme připojit, aby hlídaly případné krátkodobé přepólování na vstupních svorkách pro připojení zdroje, abychom tak zabránili poškození modulu. Nyní můžeme přistoupit k navržení schématu viz Obr. 8.1. LED indukující stav vysílače je připojena na napájecí napětí a přes ADO (pin 20) je uzemňována.



Obr. 8.2: Schéma vysílače ze ZigBit-em od firmy MeshNetics

8.1.2 DPS vysílače a osazení

Program Eagle podle zadaného schématu a knihoven vybraných součástek, vytvoří prostředí s předem zadaným rozměrem DPS a sadou součástek umístěných mimo vyznačenou oblast. Zmenšením rozměrů DPS na rozměr držáku baterie 25 x 52 mm si vyznačíme oblast nutnou pro rozmístění součástek. Jako první umístíme modul ZigBit na jednu polovinu DPS a na druhou polovinu součástky pro ovládání a svorky pro připojení baterií. Využijeme obou vrstev pro tažení vodičových cest, abychom mohli veškeré potřebné součástky umístit z horní strany desky (Top) a ty nezabraňovaly přišroubování držáku baterií k DPS spodní stranou. Proto na spodní stranu (bottom) umístíme jen napájení. Hotový návrh DPS vysílače na Obr. 8.3.



Obr. 8.3: Návrh DPS vysílače

Tab. 8.1: Seznam součástek vysílače

EAGLE Version 4.16r2 Copyright (c) 1988-2006 CadSoft

Part	Value	Device	Package	Library	Sheet
C1	220uF	C-EUC0805	C0805	rcl	1
D1		BYS10	SOD106A	telefunken	1
LED1		LED	SMARTLED-TTW	led	1
R1	10	R-EU_M0805	M0805	rcl	1
R2	10k	R-EU_M0805	M0805	rcl	1
R3	220R	R-EU_M0805	M0805	rcl	1
S1	10-XX	B3F-10XX	switch-omron	1	
S2	255SB	255SB	switch	1	
X1		MPT2	2POL254	con-phoenix-254	1
ZIGBIT1	XBEE-PRO	XBEE-PRO	XBEE-PRO	MaxStream_redukce_Zigbit	1
Dutinková lišta jednořadá 10 pinů, zlacená, RM=2.00mm					2

8.2 Přijímací zařízení

V předchozí části jsme si vytvořili jednoduchý vysílač zprostředkovávající data pro přijímací zařízení, aby byl uskutečněn celkový přenos dat a byla měřena kvalita signálu. Nyní je na řadě přijímací zařízení, které se postará o zpracování přijatých dat, zobrazení dat na LCD a komunikaci s PC. Důležitou funkcí přijímacího zařízení je schopnost pracovat již při napětí 3 V, tedy dvě alkalické mikrotužkové (AAA, R06) baterie, nebo se třemi nabíjecími bateriemi (3,6 V).

8.2.1 Schéma přijímače

Sestavíme si schéma funkčního přijímače. Pro příjem budeme používat zařízení XBee(MaxStream) a ZigBit(MeshNetics). Pro zpracování změřeného řetězce dat a ovládání dvouřádkového LCD použijí mikroprocesor MC9S08GB60, který disponuje dvěma sériovými linkami. O zajištění komunikace mezi PC a MC9S08GB60 se postará transceiver MAX3232CSE. Zobrazení naměřených dat bude dále zajištěno na LCD1602 s řadičem HD44780. Záporné napájení získáme z generátoru pulsů vytvořených pomocí 74HC04. Stabilizované napětí získáme díky MC33269D-3.3V a obsluhování procesoru zajistíme čtyřmi tlačítky. Abychom mohli využívat zařízení jako mobilní, jsou zapotřebí připojovací svorky pro připojení baterie.

Využitím mikroprocesoru MC9S08GB60 se nám nabízí možnost naprogramovat tento procesor buď za pomoci sériové linky RS232, nebo využít programovací port pro rozhraní BDM. Budeme-li chtít využívat jen programování za pomoci RS232, je nutné do procesoru nahrát bootloader, který se nám bude starat o zavedení programu na správné místo v paměti. Bohužel tento program není v MC9S08GB60 nahráný již od výrobce a je tedy nutné alespoň jednou procesor naprogramovat pomocí BDM rozhraní. Přijímač obsahuje, jak programovací rozhraní BDM, tak i RS232, a proto můžeme po prvním naprogramování přes BDM využívat RS232, pokud nemáme k dispozici programátor BDM, který je potřebný k naprogramování mikroprocesoru.

Ovládání přijímacího zařízení je umožněno pomocí čtyř tlačítek, které jsou připojena k zemi k bráně A. Port A je nastavený jako vstupní port s připojenými *PULL-UP* rezistory, které nám drží *log. 1* na vstupu, pokud není stisknuté tlačítko, které poté generuje *log. 0*. Mikroprocesor MC9S08GB60 je vybaven funkcí přerušení vyvolané klávesnicí, tedy externím zařízením připojeným na port A, mikroprocesor hlídá tlačítka při stisku pomocí vyvolaného přerušení a nemusím tento port skenovat.

Zobrazovací zařízení, v našem případě LCD1602, je připojeno pomocí 4-bitové datové komunikace a obsluhováno za pomoci dvou řídících vodičů. Celkový počet vodičů použitých pro komunikaci s LCD je šest. Zajímavým zapojením k LCD na pin Vee (ovládání kontrastu zobrazovacích krystalů) je skupina součástek vytvářející záporné napětí cca -1.5 V, které nám zvýší rozdíl napětí mezi napájecím napětím a Vee napětím. Je to z toho důvodu, že je zapotřebí vytvořit rozdíl napětí alespoň 4,5 V, aby byl vytvořen takový kontrast, jaký je nutný pro přečtení textu vypsaneho na LCD.

Indikaci stavu přijímacího zařízení zajišťují čtyři LED diody, které jsou připojené na napájecí napětí a k procesoru na port D, který je nastavený jako výstupní, a pro rozsvícení jsou diody tímto portem uzemněny. Uzemňují se z toho důvodu, že není zapotřebí z portu „ždímat“ potřebný proud, který by rozsvítil diody. Každý port má proudové omezení. Uzemňujeme-li LED diody, proud dodává zdroj, ke kterému jsou připojeny baterie nebo externí zdroj.

Stabilizátor napětí je připojený k přepínači S2, který odpojuje přijímač od zdroje. K přepínači S2 jsou připojeny i svorky určené pro baterii. Od stabilizátoru napětí jsou odděleny diodou, která zajišťuje okamžité připojení baterií v případě náhlého výpadku externího zdroje.

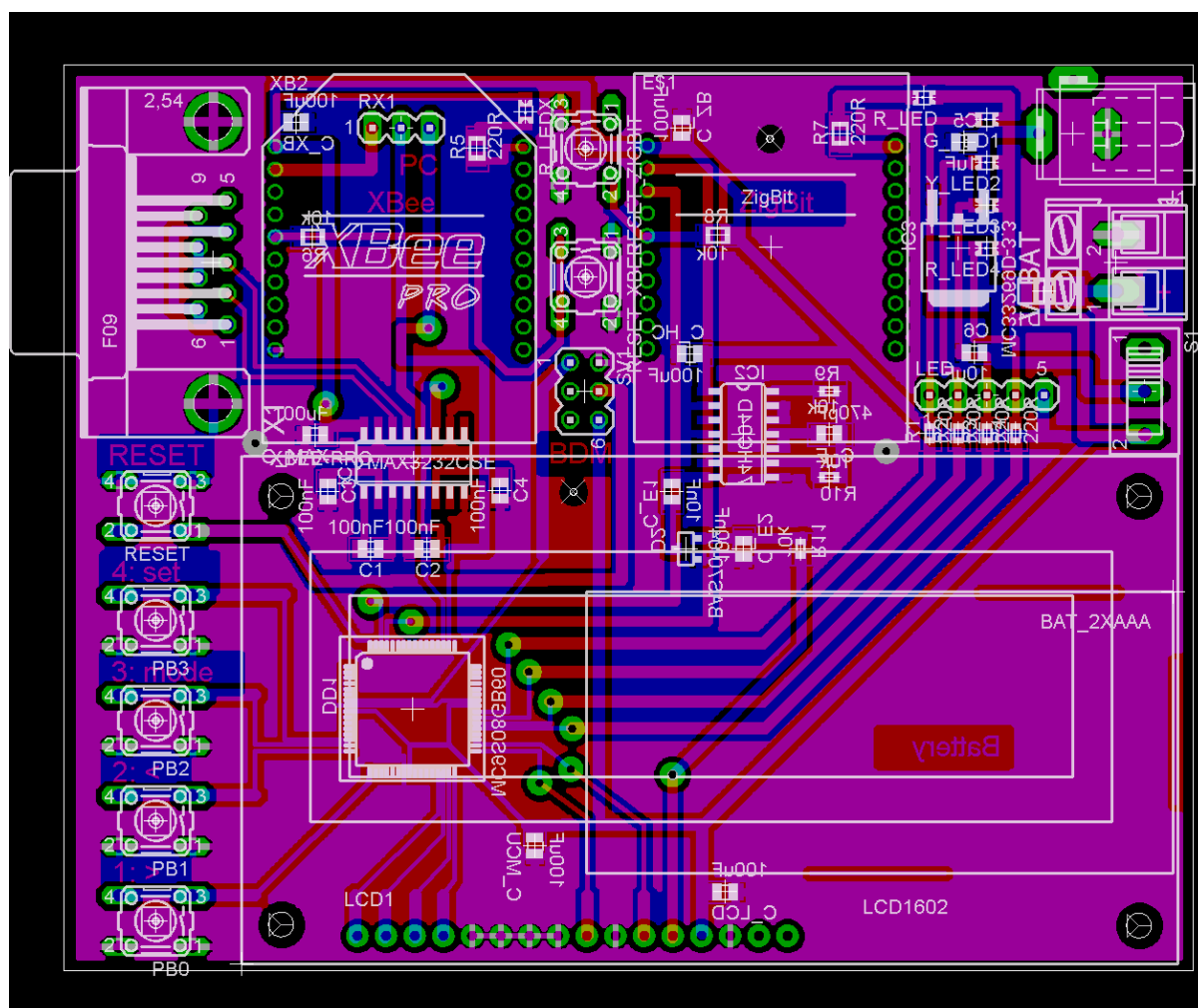
Sériovou komunikaci zajišťuje obvod MAX3232CSE, který disponuje dvěma budiči a dvěma přijímači. Přijímací zařízení využívá pouze jeden budič a přijímač, proto je zapotřebí druhý budič a druhý přijímač uzemnit, aby nedocházelo k samovolnému kmitání mezi *log. 1* a *log. 0*. viz piny 10 a 8 na MAX3232CSE. Obvod je schopný komunikovat až 120 kbps a tedy splňuje náš požadavek pro komunikaci na rychlosti 38400 bps [7].

Přijímací moduly ZigBit a XBee jsou umístěny v totožných patičkách vytvořených z dutinkové lišty 2 x 10 pinů. Jedná se o jednořadé dutinkové lišty s nízkým profilem a roztečí pinů RM = 2 mm. V kombinaci s DPS vytváří dutinkové lišty patičky pro umístění přijímacích modulů.

Přijímač obsluhovaný MC9S08GB60 se dvěma sériovými linkami umožňuje přímé připojení Tx a Rx pro modul ZigBit a kombinované připojení s PC a modulem XBee. PC lze připojit za pomoci přepnutí RX1 umístěného pod modulem XBee vpravo nahoře (viz Obr. 8.4) přepnutím do polohy PC. Tím zajistíme připojení Rx mezi PC a mikroprocesorem MC9S08GB60. Vysílání přes Tx je neustále připojené a lze nepřetržitě číst vysílaná data z mikroprocesoru. Chceme-li přepnout na přijímání dat z přijímacího modulu XBee, přehodíme jumper na RX1 do druhé pozice, a tím propojíme Rx s MC9S08GB60. V tomto módu bohužel nejde ovládat přijímací zařízení pomocí PC.

8.2.2 DPS přijímače a osazení

V návrhu DPS jsem se snažil, aby byly efektivně rozmístěné součástky z hlediska přístupu a ovládání a aby složitost plošného spoje byla co nejmenší z hlediska nutnosti vytváření prokovů. DPS je oboustranný plošný spoj o rozměrech 100 x 80 mm. Najdeme zde čtyři tlačítka pro ovládání menu přijímacího zařízení, tři resetovací tlačítka pro reset procesoru, XBee a ZigBit. Jedno LCD zobrazovací zařízení, které nám bude zobrazovat naměřená data. Po pravé straně nalezneme konektor pro připojení externího zdroje DC v rozmezí 4 – 20 V připojeného na stabilizátor napětí 3,3 V. Na svorky pro připojení baterií je možné přivést maximální stejnosměrné napájecí napětí 4 V, protože svorky jsou připojené přímo na napájení mikroprocesoru, LCD, budiče sběrnice RS232 a invertoru 74HC04. Zde **vyšší napětí než 4 V znamená zničení zařízení!** Přímo pod svorkami pro připojení baterií najdeme vypínač, který odpojuje jak baterie, tak i externí zdroj a kompletně vypne přijímací zařízení. U pravého horního rohu nalezneme LED diody, které signalizují stav přijímacího zařízení. LED diody jsou umístěné na horní straně DPS (Top layer) a je nutné vytvořit prokovy mezi Top vrstvou a Bottom vrstvou z důvodu umístění mikroprocesoru na spodní vrstvě. Po pravé straně nalezneme připojovací konektor pro sériovou linku RS232, po které můžeme komunikovat způsobem uvedeným výše. Mikroprocesor je umístěný na spodní straně plošného spoje, aby port, který zajišťuje komunikaci mezi mikroprocesorem a LCD zobrazovacím zařízením, nebyl vzájemně zrcadlen a nemusely být taženy vodivé cesty křížem přes připojovací konektor pro LCD.



Obr. 8.5: DPS přijímače s rozmístěním součástek

Uprostřed DPS těsně nad LCD je umístěné programovací rozhraní BDM, které nám umožňuje kdykoliv přeprogramovat procesor nebo z procesoru číst aktuální stav registrů a proměnných pomocí freewarové aplikace FreeMaster, která nám může sloužit při ladění programu a zároveň i efektivnímu zobrazování naměřených dat.

V blízkosti LED diod je umístěný konektor, který nám nabízí možnost připojit se k portu ovládající LED diody, což nám umožňuje externí připojení LED diod, které mohou být umístěny například přímo na krabičce zakrývající výrobek. Konektor LED obsahuje čtyři signálové vodiče a jeden napájecí vodič s hodnotou 3.3 V nebo podle napětí baterií. Na konektor LED lze připojit rovnou LED diody bez nutnosti připojení odporů. Odporů jsou umístěny na DPS.

Z důvodu nutnosti připojení baterií je DPS navržen tak, aby na spodní straně DPS bylo místo pro připojení držáku baterií. Držáku o velikosti 2 krát AAA je na spodní straně vyhrazeno místo, které je vytyčeno obdélníkem a nápisem baterie. Tímto zajistíme mobilitu výrobku a nezávislost na externím zdroji.

Ovládací tlačítka jsou vysoká, aby bylo možné výrobek uzavřít do krabice a nebyla zhoršena možnost ovládání. Připájíme-li zároveň LCD přímo na lámací lištu, budou tlačítka ve stejné výšce jako nejvyšší část LCD a v uzavřené krabičce tak bude zajištěna možnost ovládání s minimálním přesahem LCD nad krabičku, například 1 mm.

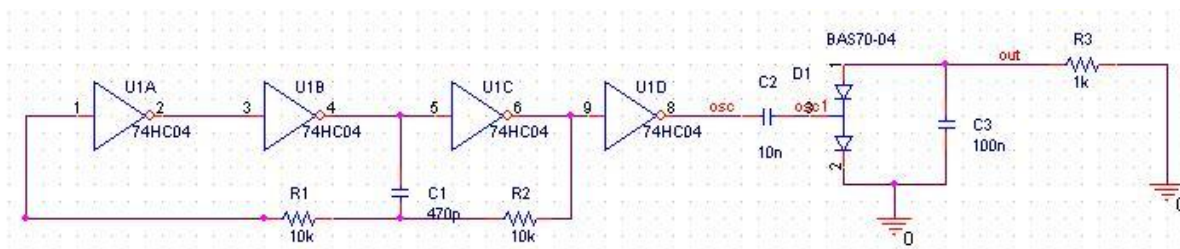
Tab. 8.2: Seznam součástek vysílače

EAGLE Version 4.16r2 Copyright (c) 1988-2006 CadSoft

Part	Value	Device	Package	Library	Sheet
C	100nF	C-EUC0805	C0805	rcl	14
C5	1uF	C-EUC0805	C0805	rcl	1
C6	10uF	C-EUC0805	C0805	rcl	1
C_E	470pF	C-EUC0805	C0805	rcl	1
C_E1	10nF	C-EUC0805	C0805	rcl	1
D1		BYS10	SOD106A	telefunken	1
D2	BAS70-04	BAS70-04	SOT23	diode	1
DD1	MC9S08GB60	MC9S08GB60	LQFP64	MC9S08GB60	1
LED		LEDCHIPLED_0805	CHIPLED_0805	led	6
IC1	MAX3232CSE	MAX3232CSE	SO16	maxim	1
IC2	74HC04D	74HC04D	SO14	74xx-eu	1
IC3	MC33269D-3.3	MC33269D-3.3	DPACK	linear	1
J1		JACK-PLUG0	SPC4077	con-jack	1
LCD1	LCD1602	LCD1602	MODUL16	LCD1602	1
PB		10-XX	B3F-10XX	switch-omron	7
R	220R	R-EU_R0603	R0603	rcl	6
R	10k	R-EU_M0805	M0805	rcl	6
RX1		MA03-1	MA03-1	con-lstb	1
S1		255SB	255SB	switch	1
SV1		MA03-2	MA03-2	con-lstb	1
V_BAT		AK300/2	AK300/2	con-ptr500	1
X1		F09HP	F09HP	con-subd	1
XB2	XBEE-PRO	XBEE-PRO	XBEE-PRO	MaxStream	1
ZIGBIT1	ZIGBIT	ZIGBIT	XBEE-PRO	MaxStream_redukce_Zigbit	1
Dutinková lišta jednořadá 10 pinů, zlacená, RM=2.00mm					4
Dutinková lišta jednořadá 20 pinů, zlacená, RM=2.54mm					1
Lámací lišta jednoradá 20p					1

8.2.3 Generování záporného napětí na Vee

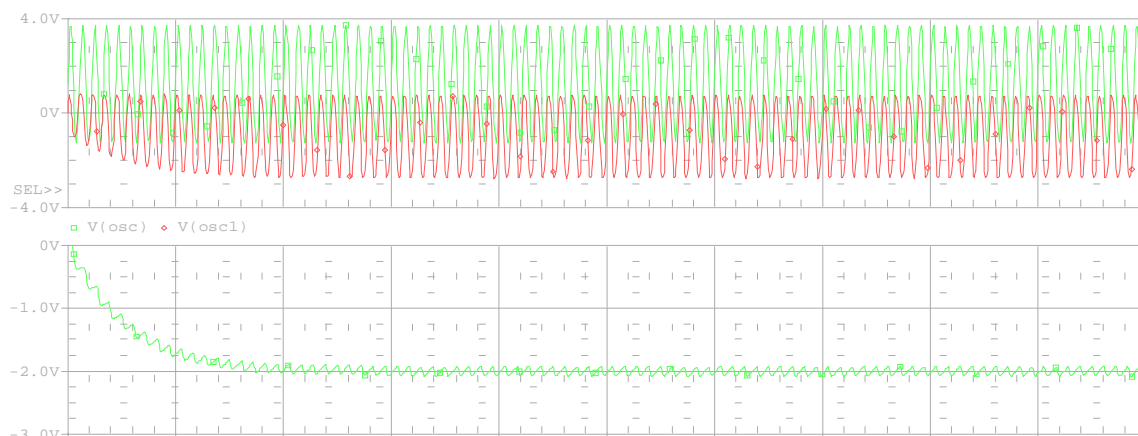
Z důvodů stanovení maximální hranice napájení na hodnotu cca 3 V a tedy možnosti napájet zařízení jenom ze dvou článků AAA, bylo proto nutné zakoupení LCD displeje, který funguje při napájecím napětí 3 V. Bohužel zakoupený displej funguje s pracovním napětím 3 V, ale s velice malým kontrastem. Z toho to důvodu bylo zapotřebí provést malou úpravu a to zvýšení rozdílu napětí stanovující kontrast displeje. Podle zapojení LCD je napájení řídicí kontrast připojené na dělič napětí, kde je zapotřebí vytvořit dostatečný rozdíl napětí pro kontrast s dobrou čitelností a to minimálně 4.5 V. Napájíme-li LCD 3 V chybí nám minimálně 1,5 V. Tuto hodnotu napětí získáme vytvořením záporného napětí vůči zemi. Abychom mohli generovat záporné napětí, je zapotřebí vyrobit generátor pulsů (viz Obr. 8.6 levá část). V našem případě obdélníkových pulsů. Generátor vytvoříme z obvodu CMOS 74HC04, což je integrovaný obvod pracující při napájecím napětí 3 V a obsahuje šest invertorů. Za pomoci čtyř invertorů vytvoříme generátor pulsů založený na zpoždění jednotlivých hradel a působení jednoho akumulárního prvku. Za pomoci odporů R1, R2 a kapacity C1 již můžeme vypočítat jejich konstantu vybíjení $\tau = C \cdot R_{R1+R2}$ a nastavit tak potřebnou frekvenci [9].



Obr. 8.6: Generátor záporného napětí

Nyní získáváme kladné kmity z $V_{P-P} = 3$ V v rozmezí 0 - 3 V. Přivedené kmity filtrujeme přes kondenzátor C2, který nám stejnosměrně oddělí generátor od druhé části obvodu. Změna napětí přivedená mezi dvě diody D1 nám usměrní napětí v propustném směru 2 a nabije nám kondenzátor. Ten se poté může vybíjet přes odpor R3 a diody do země. Protože je kondenzátor nabíjený v opačném

směru, projeví se jeho „přepólování“ na výstupu a jeví se jako záporné napětí. Obvod jsem od simuloval v programu OrCAD 10.0 (PSpice viz Obr. 8.7)



Obr. 8.7: Graf výstupního napětí Vee z oscilátoru out = $f(t)$

Generování záporného napětí se projevuje podle předpokládané teorie a dostává se až na hodnotu -2 V (viz Obr. 8.7). Vidíme, že v místě *osc1* se již hodnota napětí dostává skoro pod nulovou hranici napětí. Ve výstupním bodě *out* se hodnota napětí ustálí na -2 V a kmitá s frekvencí nastavenou na generátoru.

Sestavený obvod na výrobku dosahuje hodnoty napětí -1,7 V a frekvence 90,91 kHz při napájecím napětí $V_{cc} = 3$ V. Frekvence generátoru se liší asi o 10,6 kHz od vypočítané hodnoty, což je pravděpodobně způsobeno tolerancí hodnot použitých odporů a kondenzátoru. Podle teoretického výpočtu (4) je tato hodnota:

$$f = \frac{1}{T} = \frac{1}{t} = \frac{1}{R \times C} = \frac{1}{20 \cdot 10^3 \times 470 \cdot 10^{-12}} = 106383 \text{ Hz} . \quad (4)$$

Započítání největší možné odchylky od jmenovité hodnoty odporu 5% a kondenzátoru s tolerancí 10%: $C = 470 \cdot 10^{-12} \text{ F} = 470 \cdot 10^{-12} \times 1.1 = 517 \cdot 10^{-12} \text{ F}$ a $R = 10 \text{ k}\Omega = 10000 \times 1,05 = 10,5 \text{ k}\Omega$

$$f = \frac{1}{T} = \frac{1}{t} = \frac{1}{R \times C} = \frac{1}{2 \times 10,5 \cdot 10^3 \times 517 \cdot 10^{-12}} = 92106 \text{ Hz} . \quad (5)$$

Při započtení tolerancí jsme se dostali na hodnotu cca 92,1 KHz (viz (5)), kde vzniká rozdíl mezi teoretickou a změřenou hodnotou pomocí osciloskopu HP 54621A (viz Obr. 8.8) $\Delta f = 1,196 \text{ kHz}$.



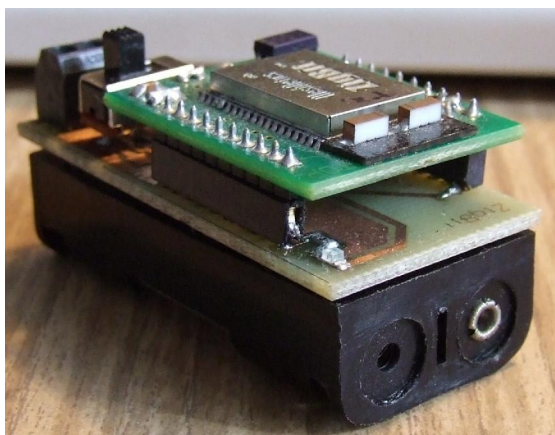
Obr. 8.8: Změřená frekvence na sestaveném generátoru v přijímacím zařízení

Výstup generátoru napětí připojím přímo na vstup LCD a pin Vee. Nyní bude změřené napětí mezi V_{cc} a Vee rovno hodnotě -4,7 V. Takový rozdíl napětí nám způsobí vysoký kontrast na LCD a dobrou čitelnost znaků vypsáných na displeji.

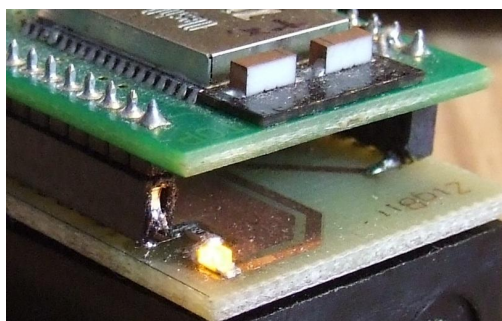
8.3 Zařízení v praxi

8.3.1 Vysílací zařízení

Malé vysílací zařízení poskytuje vysokou mobilnost a možnost snadného umístění na požadované místo. Držák na dvě mikrotužkové (AAA) baterie stanovil rozměry DPS 25 x 52 mm s výškou zkompletovaného vysílače 25 mm. Na Obr. 8.9 vidíme dutinkovou lištu, která umožňuje vyjmout modul. Jedna polovina DPS zabraná modulem ZigBit od firmy MeshNetics je osazena LED diodou indikující aktivitu zařízení viz Obr. 8.10.

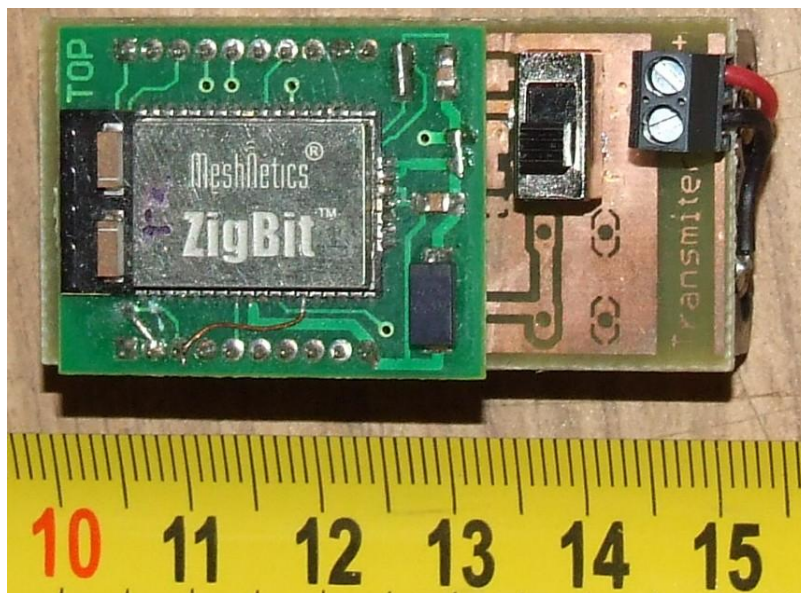


Obr. 8.9: Vysílací zařízení



Obr. 8.10: LED indikující aktivní vysílač

Vidíme držák baterií ve stejné velikosti jako DPS, který je na držáku baterií posazený a uprostřed držáku přichycený šroubky pro pevné spojení.

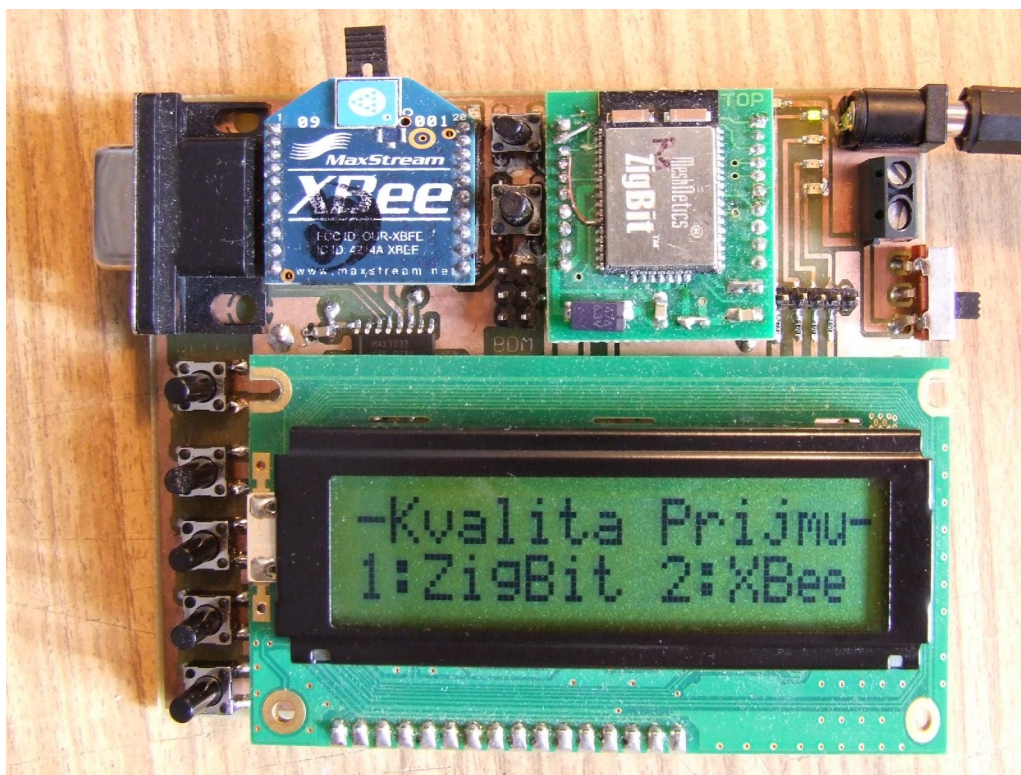


Obr. 8.11: Horní pohled na vysílač

Horní pohled na vysílač poskytuje Obr. 8.11, kde je krásně vidět redukce ze ZigBit na XBee verzi rozmístění konektorů, vyrobenou firmou BetaControl. Vypínač odpojuje připojené baterie a tím šetří baterie proti zbytečnému vybíjení. Hned vedle vypínače je možnost umístit RESETovací tlačítko. Ochranná dioda s odporem proti náhodnému krátkodobému přepólování je překryta redukcí ZigBit a XBee. Svorky na baterii vytvářejí rozebíratelný spoj a je možnost připojit jiný zdroj. Napájecí napětí je v rozsahu 1.8 – 3.6 V.

8.3.2 Přijímací zařízení

Narozdíl od vysílacího zařízení, kde bylo zapotřebí umístit jen jeden modul ZigBit a malé doplňky je u přijímacího zařízení využito dvou modulů, zobrazovací zařízení LCD pro efektivní poskytnutí naměřených výsledků, dále možnost připojení externího zdroje a baterie a umístění ovládání a zprostředkování propojení mezi PC a přijímacím zařízením viz Obr. 8.12, kde vidíme osazený DPS.



Obr. 8.12: Výsledný komplet přijímacího zařízení

Viditelně největší plochu zabírá zobrazovací zařízení, které může být upevněné na DPS pomocí jednořadé šestnácti pinové lámací lišty a to buďto napevno nerozebíratelně připájeno, nebo za pomoci dutinkové lišty nasunutou na lámací lištu připíjenou k DPS. Zajištění LCD provedeme pomocí čtyř distančních sloupků pro lepší mechanickou pevnost. Upevníme-li LCD k DPS připájením, můžeme snížit výšku přijímacího zařízení a tím zajistit snadnou obsluhu zařízení pomocí tlačítek v případě, že bychom umístili zařízení do krabičky.

Přijímací zařízení můžeme napájet, jak z externího zdroje, tak z baterií připojených ke svorkám *X_bat* viz Obr. 8.12. Po přivedení napájecího napětí do přijímacího zařízení jej můžeme zapnout pomocí přepínače umístěného na pravé straně hned pod svorkami pro připojení baterií *prep1*. Přepínač propojí zdroj s napájecími obvody celého přijímacího zařízení a dojde k oživení všech součástí.

Přivedené napájecí napětí aktivuje procesor MC9S08GB60, který se z inicializuje a jako první viditelnou činnost rozsvítí LED1, která signalizuje spuštění přijímacího zařízení a zároveň funkčnost procesoru.

Další inicializace LCD nastaví displej do požadovaného režimu a sdělí uživateli jaká aplikace bude spuštěna viz Obr. 8.13.



Obr. 8.13: Úvodní informační zpráva

Displej zobrazující naměřená data je maticový dvouřádkový displej o šestnácti znacích na řádek. Po krátkém intervalu zobrazení úvodní zprávy se přepíše informace o verzi softwaru, který je nahraný v mikroprocesoru obsluhujícím displej.



Obr. 8.14: Informace o verzi softwaru pro zobrazování

Následuje první důležitá informace pro uživatele, kterému je nabídnuta možnost zvolit si k měření kvality přijímaného signálu jeden ze dvou modulů. Zobrazená zpráva formou textu vyzývá k stisknutí tlačítka č. 1 pro výběr modulu ZigBit nebo tlačítka č. 2 k výběru XBee.



Obr. 8.15: Volba modulu

Zobrazení je rozdělené do dvou částí. První část a tedy i první řádek slouží k rozlišení přijímacího zařízení, ze kterého získáváme naměřená data viz Obr. 8.16 a snadno rozlišíme zdroj dat.



a)



b)

Obr. 8.16: a) naměřená data ze ZigBit

b) naměřené data z XBee

Vytvořené menu obsahuje celkem pět položek ve kterých můžeme dále listovat za pomoci tlačítka č. 1 (>) a tlačítka č. 2 (<). Zobrazené jednociferné číslo na začátku druhého řádku číselně značí položku v menu. Nejnížší položkou v menu FC (Frame Counter) zobrazující číslo rámce. Vysílač neustále vysílá jeden rámec za druhým s časovou prodlevou mezi rámci 50 ms, ale naměřené informace jsou aktualizovány na displeji dvakrát za vteřinu. Přijímá-li přijímací modul data, je vypočítávána kvalita signálu a FC je neustále inkrementován. Dojde-li ke ztrátě signálu, zastaví se zvyšování počtu přijatých rámců.

Následující položka z menu označená pod číslem 3 zobrazuje počet chybných rámců *FEC(Frame Error Counter)* viz Obr. 8.17.



a)



b)

Obr. 8.17: *FEC(Frame Error Counter)* Počet chybných rámců pro a) ZigBee b) XBee

Celkový počet chybných bitů v přenesených rámcích je označována jako *BER (Bite Error Counter)*.



a)



b)

Obr. 8.18: *BEC(Bit Error Counter)* Počet chybných bitů v rámcích pro a) ZigBee b) XBee

Označení číslo pět nese ukazatel kvality rámců na přenosové trase, neboli tzv. *LQI (Link Quality Indicator)*



a)



b)

Obr. 8.19: *LQI(Link Quality Indicator)* pro a) ZigBee b) XBee

Jako poslední a v podstatě nejdůležitější hodnota je *RSSI(Received Signal Strength Indicator)*, která nám udává útlum přijímaného signálu. Již z toho parametru můžeme zjistit, je-li možnost použít danou vzdálenost mezi vysílačem a přijímačem. Jednotka zobrazované hodnoty je uvedena v dBm viz Obr. 8.20 a její údaj, jako všechny parametry posílané přes UART jsou aktualizované po 0,5 s. Z hlediska jednoduššího odčítání hodnoty zobrazované na displeji jsem využil možností tzv. *BAR* grafu, který graficky zobrazuje úroveň útlumu přijímaného signálu viz Obr. 8.21.

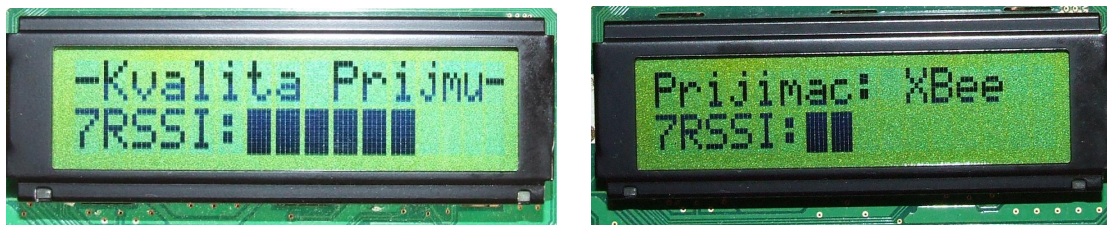


a)



b)

Obr. 8.20: *RSSI(Received Signal Strength Indicator)* pro a) ZigBee b) XBee

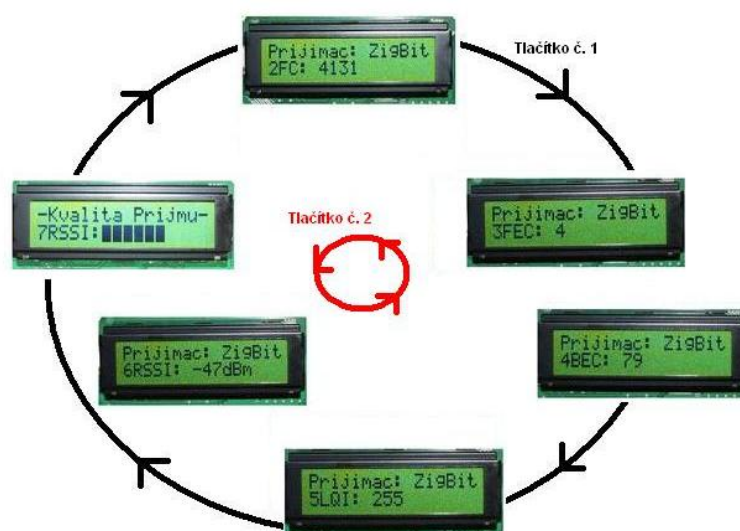


a)

b)

Obr. 8.21: RSSI(Received Signal Strength Indicator) BAR graf pro a) ZigBee b) XBee

BAR graf zastupuje hodnotu naměřenou a zobrazenou viz Obr. 8.20. Bude-li hodnota RSSI = 0 dBm bude bar graf ukazovat plnou výchylku a tedy plný řádek obdélníků. Na rozdíl při zobrazování nižší úrovně než -99 dBm bude řádek prázdný a výchylka BAR grafu směrem doprava znamená kvalitnější přijímaný signál a opačně.



Obr. 8.22: Přehled prohlížení menu

Přijímací zařízení nabízí čtyři ovládací tlačítka. První dvě tlačítka jsou určené pro ovládání menu viz předchozí část. Dále nám tady zůstávají dvě volná ovládací tlačítka, která můžeme využít pro ovládání změny vybraného zařízení. Tlačítko č. 3 vyhradíme pro změnu zařízení na ZigBit a tlačítko č. 4 využijeme pro změnu získávání dat z přijímacího modulu XBee.

Další možností změny přijímaných dat ze ZigBit nebo XBee můžeme vyřešit pátým tlačítkem umístěným na přijímacím zařízení a sloužícím pro možnost RESETu procesoru ovládajícího LCD a zpracovává naměřená data.

9 Měření v praxi



9.1 Odběr elektrické energie

9.1.1 Přijímací zařízení

Po sestavení přijímacího zařízení a následném oživení máme nyní možnost zjistit odběr proudu během provozu, jak při špatné kvalitě přijímaného signálu, tak i při velmi dobrém příjmu signálu. Proto jediná vlastnost, která se může během měření měnit je odběr proudu přijímacího modulu a počet připojených modulů na přijímacím zařízení.

Tab. 9.1: Teoretický odběr proudu podle jmenovitých hodnot udaných výrobcem(přijímač)

Zařízení	I_{IN_teor}		
	typ	max	jednotka
XBee	50	50	mA
ZigBit	19	19	mA
MAX3232CSE	0.3	1	mA
MC9S08GB60	7	120	mA
74HC04	-	100	mA
PC 1602-L	2	3	mA
Celkem	58	293	mA

Podle hodnot udávaných výrobcí by se průměrná hodnota odebíraného proudu při napájecím napětí $U_{CC} = 3.3 \text{ V}$ měla pohybovat okolo $I_{IN_teor} = 58 \text{ mA}$ a s teoreticky maximálním $I_{IN_teor} = 293 \text{ mA}$

Tab. 9.2: Změřený odběr přijímacího zařízení

Kvalita přijímaného signálu	žádný signál		slabý signál		silný signál		jednotka
Zařízení	Se stabilizátorem	přes bateriové svorky	Se stabilizátorem	přes bateriové svorky	Se stabilizátorem	přes bateriové svorky	
Přijímací zařízení	I_{INstab}	I_{INbat}	I_{INstab}	I_{INbat}	I_{INstab}	I_{INbat}	
ZigBit a XBee	84 (0.277)	77 (0.254)	-88 dBm 79 (0.260)	-83 dBm 80 (0.264)	-7 dBm 78 (0.257)	-7 dBm 73 (0.240)	mA
bez XBee	40 (0.132)	44 (0.145)	-85 dBm 37 (0.122)	36 (0.118)	-7 dBm 39 (0.128)	-7 dBm 37 (0.122)	mA
bez ZigBit	57 (0.188)	60 (0.198)	-87 dBm 57 (0.188)	-95 dBm 60 (0.198)	-75dBm 57 (0.188)	-77 dBm 55 (0.181)	mA
Bez ZigBit a XBee	15 (0.049)	14 (0.046)					mA

(P [W]- příkon přijímacího zařízení)

Výsledky z měření odběru proudu a tedy i získání příkonu přijímacího zařízení (červeně v závorkách viz Tab. 9.2) nám ukázal skutečný proud odebíraný při napájecím napětí $U_{IN} = 3.3 \text{ V}$. Nevětší odběr je logicky při plném osazení přijímacího zařízení oběma moduly a $I_{IN} = 84 \text{ mA}$. Teoreticky vypočítaná hodnota vstupního proudu byla $I_{IN_teor} = 58 \text{ mA}$. Na větším příkonu se pravděpodobně podílejí nezapočítaný invertor 74HC04 a mikroprocesor, který při zpracovávání řetěze a řízení LCD odebírá vyšší proud než $I_{proc} = 7 \text{ mA}$.

Vysílací zařízení odebírá při plném osazení proud $I_{IN} = 84 \text{ mA}$, který nám zajistí dobu provozu při připojení stejných baterií jako na vysílacím zapojení přibližně 643 min viz (6).

$$t_{\text{výdrž}_R} = \frac{\text{Kapacita}_{\text{baterie}}}{I_{\text{vybíjecí}}} \times 60 = \frac{900 \text{ mAh}}{84 \text{ mA}} \times 60 = 643 \text{ min} \quad (6)$$

Protože je zde nutné při měření kvality přenosové trasy, aby přijímací zařízení vydrželo minimálně stejnou dobu provozu jako vysílací zařízení, musíme zvýšit kapacitu napájecích baterií pro přijímací zařízení, není-li samozřejmě napájené z externího zdroje s dobou napájení $t_{\text{extern}} = \infty$. Minimální dobu provozu přijímacího zařízení jsi stanovíme $t_{\text{výdrž}_T} = t_{\text{výdrž}_R}$ a kapacitu akumulátorů dopočítáme viz (7).

$$\text{Kapacita}_{\text{baterie}_R} = \frac{t_{\text{výdrž}_T} \times I_{\text{vybíjecí}}}{60} = \frac{2842 \text{ min} \times 84 \text{ mA}}{60} = 3978 \text{ mAh} \quad (7)$$

Budeme-li předpokládat, že provoz přijímacího zařízení bude pouze s modulem ZigBit od MeshNetics, bude odebíraný proud dosahovat hodnoty $I_{IN} = 40 \text{ mA}$ a tím bude snížen odběr proudu na polovinu. Nyní bude potřebná kapacita baterií $\text{Kapacita}_{\text{baterie}_{R_ZigBit}} = 1894 \text{ min}$ viz (8).

$$\text{Kapacita}_{\text{baterie}_{R_ZigBit}} = \frac{t_{\text{výdrž}_T} \times I_{\text{vybíjecí}}}{60} = \frac{2842 \text{ min} \times 40 \text{ mA}}{60} = 1894 \text{ mAh} \quad (8)$$

Zjištěná nejnižší **kapacita baterií pro přijímací zařízení** jen se ZigBit modulem je **2100 mAh** za použití tužkových baterií (AA , R06) a zajistíme tak delší dobu provozu přijímacího zařízení cca 2,2 dny.

9.1.2 Vysílací zařízení

Vysílací zařízení tvoří pouze zdroj z baterií a vysílací modul ZigBit. Základní připojení vysílacího modulu VCC, GND, RESET a jednoho I/O pinu pro signalizaci aktivního stavu se bude průměrná hodnota proudového odběru pohybovat kolem hodnoty proudu vysílacího modulu udané výrobcem.

Tab. 9.3: Teoretický odběr proudu podle jmenovitých hodnot (vysílač)

Zařízení	I_{IN_teor}		
	typ	max	jednotka
ZigBit	19	19	mA
R_RESET	0.24	0.24	mA
Celkem	19.24	19.24	mA

Vysílací zařízení po zapnutí neustále vysílá sekvenci dat a proto je vysílací zařízení neustále v režimu vysílání, proto by se odebíraný proud měl pohybovat blízko hranice $I_{CC} = 19 \text{ mA}$ udané výrobcem.

Tab. 9.4: Změřený odběr vysílacího zařízení

Zařízení	I_{IN}	jednotka
Vysílací	19 (0.062)	mA
(P [W] – příkon zařízení)		

Praktické měření nám potvrdilo teoretickou domněnku odebíraného proudu při napájecím napětí $U_{IN} = 3,3 \text{ V}$, že maximální odbíraný proud $I_{IN} = 19 \text{ mA}$ tedy $I_{IN} = I_{IN_teor}$.

Nyní můžeme vypočítat pravděpodobnou výdrž vysílacího zařízení napájeného dvěma mikrotužkovými bateriemi (AAA, R03) s kapacitou 900 mAh.

$$t_{\text{výdrž}_T} = \frac{\text{Kapacita}_{\text{baterie}}}{I_{\text{vybíjecí}}} \times 60 = \frac{900 \text{ mAh}}{19 \text{ mA}} \times 60 = 2842 \text{ min} \quad (9)$$

Výsledek **doby provozu na baterie** je $t_{\text{výdrž}_T} = \mathbf{2842 \text{ minut}}$ tedy 1.97 dne s (**AAA, R03**) **kapacitou 900 mAh**. Po tuto dobu je možné měřit kvalitu přenosové cesty mezi přijímačem a vysílačem. Až se vysílač vybije na přijímacím zařízení bude zobrazena ztráta signálu a bude si pamatovat poslední naměřené údaje jako je FC, FEC a BER, které budou pro nás v tom to případě nejdůležitější.

9.2 Shrnutí

Použité hodnoty proudů jsou brány pro případ největšího vytížení **à** bez signálu viz Tab. 9.2 **à** největším odběr proudů. Výsledky zjištěné pomocí měření a výpočtů jsou uvedeny v Tab. 9.5.

Přijímací zařízení Vysílací zařízení

Tab. 9.5: Napájení

Napájení	baterie	p	p
	externí	p	..
Počet baterií		3 x (AA, R06)	2 x (AAA, R03)
Kapacita článku		2100 mAh	900 mAh
Doba provozu	jen ZigBit	3150 minut	2842 minut
	jen XBee	2210 minut	-
	ZigBit a XBee	1500 minut	-
Změřená doba provozu	ZigBit a XBee	720 minut	2850 minut

Zařízení jsem uvedl do provozu a nechal spuštěné cca 17hodin. Vzdálenost mezi přijímačem a vysílačem byla malá cca 70 cm. Přijímací zařízení mělo hned vedle sebe umístěné zařízení Bluetooth cca 25 cm a NB s Wi-fi s občasným připojením ve vzdálenosti přibližně 20 cm viz Obr. 9.1. Toto byly nejsilnější zdroje, ale v okolí byly aktivní minimálně čtyři další Wi-Fi sítě.

Wi-Fi D R = 20 cm



Bluetooth D R = 25 cm



Vzdálenosti T D R = 70cm m



Obr. 9.1: 17-ti hodinový praktický test

Výsledné měření bylo ukončené z důvodů vybití baterií na vysílacím zařízení. Vybíjení baterií až do úplného vybití mohlo ovlivnit výkon vysílače a tedy i počet špatně přenesených bitů. Výsledek měření je zkrácen do posledního řádku naměřeného přijímacími moduly, kde zachycená zpráva přes SCI sériové rozhraní je uložena pomocí programu Tera Term viz Obr. 9.2. a zaznamenán v Tab. 9.6. Další praktický test využití zařízení najdeme v Příloha E.



Obr. 9.2: Výsledek 17-ti hodinového praktického testu

Tab. 9.6: Výsledek 17-ti hodinového praktického testu bez Bluetooth, bez Wi-fi

```

*****Kvalita Prijmu*****
*** Ridici MCU ***
-Hlaseni po SCI-

Inicilizace MCU.....Hotovo
Inicilizace LCD ....Hotovo

-Kvalita Prijmu-
ZigBit x XBee
Ver.: 1.00 VSMB
Vybrany prijmac: ZigBit

FC=5356724      FEC=56      BEC=27621      LQI=212      RSSI=-61dBm(09)

Vybrany prijmac: XBee

FC=9784      FEC=9784      BEC=2392732      LQI=255      RSSI=-95dBm (190)

```

Tab. 9.7: Výsledek druhého 17-ti hodinového praktického testu a náhodné puštění Bluetooth a Wi-fi

```

Vybrany prijmac: ZigBit

FC=5363110      FEC=945      BEC=64691      LQI=254      RSSI=-63dBm(08)

Vybrany prijmac: XBee

FC=9259      FEC=9259      BEC=2283203      LQI=255      RSSI=-95dBm (190)

```

Závěr

Realizace nástroje pro projektování a instalaci radiových sítí podle standardu IEEE802.14.5 dopadla následovně. **Vysílací zařízení** viz kapitola 8.3.1 Vysílací zařízení je realizováno jako malé vysílací zařízení s vysokou mobilností. Rozměry vysílacího zařízení jsou stanovené podle držáku baterií na rozměry 25 x 52 mm a výškou zkompletovaného vysílače 25 mm. Vysílač je osazený modulem ZigBit od firmy MeshNetics s proudovým odběrem 19 mA při napájecím napětí 3,3 V. Vysílač napájí dvě nabíjecí mikrotužkové baterie (AAA, R03) s kapacitou 900 mA, které nám poskytují dobu vysílání až 2842 minut viz kapitola 9.2 Shrnutí Tab. 9.5: Napájení. V kapitole 9.2 Shrnutí je prakticky ověřená doba provozu na baterie, jak vysílacího zařízení, tak i přijímacího zařízení. Přijímací zařízení potvrdilo teoretický výpočet a dokonce překonalo teoretickou dobu provozu o 8 minut. Vysílací zařízení indikuje svoji činnost periodickým bliknutím $T = 4$ s LED diody umístěné na DPS vysílacího zařízení.

Vyrobené **přijímací zařízení** o rozměrech 100 x 80 mm je o něco větší než vysílací zařízení, ale pořád dostatečně mobilní. Pro jeho dobrou mobilitu, je možné přijímací zařízení napájet buď z externího zdroje nebo z baterií. Napětí připojeného externího zdroje musí být stejnosměrné v rozsahu 4,55 V – 20 V. Při přivedení napájecího napětí na bateriové svorky se musíme vyvarovat přivedení napětí vyššího než 4 V, abychom nezničili připojená zařízení. Konstrukce přijímacího zařízení je provedena s co nejmenšími rozměry podle druhu použitých součástek. Proudový odběr přijímacího zařízení při plném osazení je 84 mA, kde podle teoretického výpočtu při připojení baterií s kapacitou 2100 mAh dosáhneme výdrže 1500 minut viz 9.2 Shrnutí. Test s připojenými bateriemi splnil necelou polovinu provozní doby a to 720 minut. Pravděpodobná příčina poloviční výdrže je způsobena použitím nekvalitních nabíjecích baterií, které mají za sebou dvouletou životnost v digitálním fotoaparátu a jsou nabíjené rychlonabíječkou. Obsluha přijímacího zařízení je velmi jednoduchá a přehledná viz kapitola 8.3.2 Přijímací zařízení. Ihned po zapnutí moduly začnou okamžitě přijímat data, bude-li spuštěno vysílací zařízení. Mikroprocesor MC9S08GB60 určený pro obsluhu LCD a zpracování naměřených dat nám po nastartování vypisuje úvodní texty na LCD a zastaví se na volbě vybrání modulu, ze kterého chceme zpracovávat naměřená data. Pomocí ovládacích tlačítek si zvolíme modul a následuje periodické zobrazování naměřených dat aktualizovaných po 0,5 s. Budeme-li sledovat hodnotu FC (Frame Counter) zjistíme, že vysílač vysílá pseudonáhodnou posloupnost každých 10 ms a následné zobrazení FC nám roste přibližně o hodnotu 53. Zobrazená hodnota FEC (Frame Error Counter) nám počítá každý špatný rámec, který obsahoval alespoň jeden chybně přenesený bit. Tímto se dostáváme k hodnotě BEC, která nám určuje každý chybně přenesený bit od začátku měření, tedy zapnutí přijímacího zařízení. Přijímací zařízení nabízí i měření kvality linky LQI (Link Quality Indikator). LQI zobrazuje maximální hodnotu 255 a začne se snižovat až v případě velice velkého zhoršení kvality signálu signálu \hat{a} při výskytu velkého množství chybných bitů v přeneseném rámci. Osobně jsem si vyzkoušel několik měření. V kapitole 9.2 Shrnutí Tab. 9.6 a Tab. 9.7 jsou naměřené hodnoty z praktického měření na vzdálenost 70 cm a rušením i nerušením ostatních mobilních zařízení pracujících na stejné frekvenci 2,4 GHz. Měřená vzdálenost není nijak velká, což potvrzuje i velikost FEC = 56, což znamená, že v 56 přenesených rámcích se objevila alespoň jedna chyba. Vypočítaná chybovost BER za dobu 17 hodin při celkovém počtu přenesených rámců je $5 \cdot 10^{-6}$ bez náhodného rušení okolních bezdrátových zařízení umístěných v těsné blízkosti přijímacího zařízení. Pro lepší představitelnost jsem udělal další testy zaměřené na možnosti rozmístění senzorů v bytě a pravděpodobně maximální vzdálenosti použití modulu ZigBit viz Příloha E. Změřené body ukazují i přes malou hodnotu kvality signálu na přenos s malým procentem výskytu chyb. Jediný bod, kdy jsem naměřil několik špatně přenesených bitů je v bodě měření 4. Maximální změřená vzdálenost v otevřeném prostoru pro přenos dat byla 154 m. V této vzdálenosti byl stále signál umožňující příjem dat. V přijímacím zařízení je možné provádět řadu úprav, které jsou spojeny se zpracováním naměřených dat. Přijímací zařízení může být rozšířeno např. o průměrování naměřených hodnot LQI, které se během měření mění, a každá naměřená hodnota je právě aktuální k danému rámci. Další možností je sledování FC a po dosažení požadované hodnoty mohou být naměřená data uložena do paměti pod konkrétním měřením. Poté by mohlo být zapnuto vyčtení

naměřených hodnot na požadovaném místě a připojení přijímacího zařízení k PC. Jedním z užitečných řešení, může být vytvoření aplikace přímo na PC, která bude určena přímo k získávání dat z přijímacího zařízení a umožní nám tak jednoduše a hlavně rychle ukládat naměřená data například do Excelu pro další zpracování.

Literatura

- [1] S08GB Produkt Summary Page [online]. 2008 , 30.4.2008 [cit. 2008-01-01]. Dostupný z WWW:
<http://www.freescale.com/files/microcontrollers/doc/data_sheet/MC9S08GB60A.pdf?fp=1&WT_TYPE=Data%20Sheets&WT_VENDOR=FREESCALE&WT_FILE_FORMAT=pdf&WT_ASSET=Documentation>
- [2] HD44780U (LCD-II) : (Dot Matrix Liquid Crystal Displej Controller/Driver) [online]. 1998 [cit. 2008-04-30]. Dostupný z WWW:
<<http://www.sparkfun.com/datasheets/LCD/HD44780.pdf>>.
- [3] How to control a HD44780-based Character-LCD : The Industry Standard Character LCD [online]. 1996 , 2005-09-20 [cit. 2008-04-30]. Dostupný z WWW:
<http://home.iae.nl/users/pouwaha/lcd/lcd0.shtml#instruction_set>.
- [4] CadSoft Online [online]. 2008 , 13.5.2008 [cit. 2008-01-01] Dostupný z WWW:
<<http://www.cadsoft.de/freeware.htm>>
- [5] Eagle – INSTALUJ.cz [online]. 2008, 13.5.2008 [cit. 1.1.2008] Dostupný z WWW:
<<http://www.instaluj.cz/cz/katalog/domacnost-zabava/hobby/eagle/>>
- [6] DataSheet PDF BYS10 [online]. 2008, 13.5.2008, [cit. 08.06.2004]. Dostupný z WWW:
<<http://www.datasheetcatalog.org/datasheet/vishay/bys10.pdf>>
- [7] DataSheet PDF RS-232 [online]. 2008, 13.5.2008, [cit. 1.03.1999]. Dostupný z WWW:
<<http://gme.cz/dokumentace/dokumenty/959/959-250/dsh.959-250.1.pdf>>
- [8] ZigBit Module with Dual Chip Antenna [online]. 2008 , 13.5.2008 [cit. 2008-01-01]. Dostupný z WWW: <<http://MeshNetics.com/zigbee-modules/chip-antenna/>>
- [9] ELM – Using an LCD at low supply voltage [online]. 2008, 10.4.2008 [cit. 1.1.2008]. Dostupný z WWW: <<http://elm-chan.org/docs/lcd/lcd3v.html>>
- [10] Wikipedie, otevřená encyklopedie [online]. 2008, 30.4.2007 [cit. 1.1.2007] Dostupný z WWW: <<http://cs.wikipedia.org/wiki/ZigBee>>
- [11] MeshNetics OpenMAC [online]. 2008 , 13.5.2008 [cit. 01.01.2008]. Dostupný z WWW:
<<http://MeshNetics.com/opensource/mac/>>
- [12] SKLÁDANÝ, V. Ultrazvukový měřič vzdálenosti. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2006. 43 s. Vedoucí bakalářské práce Ing. Zbyněk Fedra.
- [13] PATRICK, D. Wireless Network Coexistence. Englewood Cliffs: McGraw-Hill Professional Publishing, 2004. ISBN 0071399151
- [14] GUTIERREZ, CALLAWAY, BARRETT. Enabling Wireless Sensors with IEEE802.15.4, Low-Rate Wireless Personal Area Networks. Piscataway: IEEE Press, 2003. ISBN 0738135577
- [15] Aplikační zprávy firmy Freescale (www.freescale.com)
- [16] ZigBee Alliance [online]. 2008 , 13.5.2008 [cit. 01.01.2008]. Dostupný z WWW: <<http://www.zigbee.org/en/index.asp>>
- [17] Dokumentation – XBee® ZNet 2.5 Mesh RF Modules – Digi International [online]. 2008, 13.5.2008 [cit. 01.01.2008]. Dostupný z WWW:
<<http://www.digi.com/products/wireless/zigbee-mesh/xbee-series2-moduledocs.jsp>>
- [18] PC1602-L Datasheet pdf – OUTLINE DIMENSION & BLOCK DIAGRAM – Power Technology [online]. 2008, 15.5.2008 [cit. 01.01.2008]. Dostupný z WWW:
<http://www.datasheetcatalog.com/datasheets_pdf/P/C/1/6/PC1602-L.shtml>

Seznam symbolů

I_{IN}	vstupní proud do přijímacího/vysílacího zařízení
I_{INbat}	vstupní proud do přijímacího zařízení přes bateriové svorky
I_{INstab}	vstupní proud do přijímacího zařízení přes konektor externího napájení
$I_{vybíjecí}$	proud zatěžující přijímací/vysílací zařízení při provozu
$Kapacita_baterie_R$	kapacita baterií pro přijímací zařízení při plném osazení
$Kapacita_baterie_{R_ZigBit}$	kapacita baterií pro přijímacího zařízení potřebná s připojeným ZigBit modulem
$t_{výdrž_T}$	doba výdrže vysílacího zařízení při provozu na baterie
$t_{výdrž_R}$	doba výdrže přijímacího zařízení při provozu na baterie

Seznam zkratek

BEC	Bite Error Counter
FC	Frame Counter
FEC	Frame Error Counter
LQI	Link Quality Indicator
RSSI	Received Signal Strength Indicator

Přílohy

Příloha A

Tab. A: Instrukce pro řadič HD44780

Instruction	Code										Description	Execution time**
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear display	0	0	0	0	0	0	0	0	0	1	Clears display and returns cursor to the home position (address 0).	1.64mS
Cursor home	0	0	0	0	0	0	0	0	1	*	Returns cursor to home position (address 0). Also returns display being shifted to the original position. DDRAM contents remains unchanged.	1.64mS
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction (I/D), specifies to shift the display (S). These operations are performed during data read/write.	40uS
Display On/Off control	0	0	0	0	0	0	1	D	C	B	Sets On/Off of all display (D), cursor On/Off (C) and blink of cursor position character (B).	40uS
Cursor/display shift	0	0	0	0	0	1	S/C	R/L	*	*	Sets cursor-move or display-shift (S/C), shift direction (R/L). DDRAM contents remains unchanged.	40uS
Function set	0	0	0	0	1	DL	N	F	*	*	Sets interface data length (DL), number of display line (N) and character font(F).	40uS
Set CGRAM address	0	0	0	1	CGRAM address						Sets the CGRAM address. CGRAM data is sent and received after this setting.	40uS
Set DDRAM address	0	0	1		DDRAM address						Sets the DDRAM address. DDRAM data is sent and received after this setting.	40uS
Read busy-flag and address counter	0	1	BF		CGRAM / DDRAM address						Reads Busy-flag (BF) indicating internal operation is being performed and reads CGRAM or DDRAM address counter contents (depending on previous instruction).	0uS
Write to CGRAM or DDRAM	1	0			write data						Writes data to CGRAM or DDRAM.	40uS
Read from CGRAM or DDRAM	1	1			read data						Reads data from CGRAM or DDRAM.	40uS

Remarks:

- DDRAM = Display Data RAM.
- CGRAM = Character Generator RAM.
- DDRAM address corresponds to cursor position.
- * = Don't care.
- ** = Based on $F_{osc} = 250\text{kHz}$.

Table 2.4. Bit names

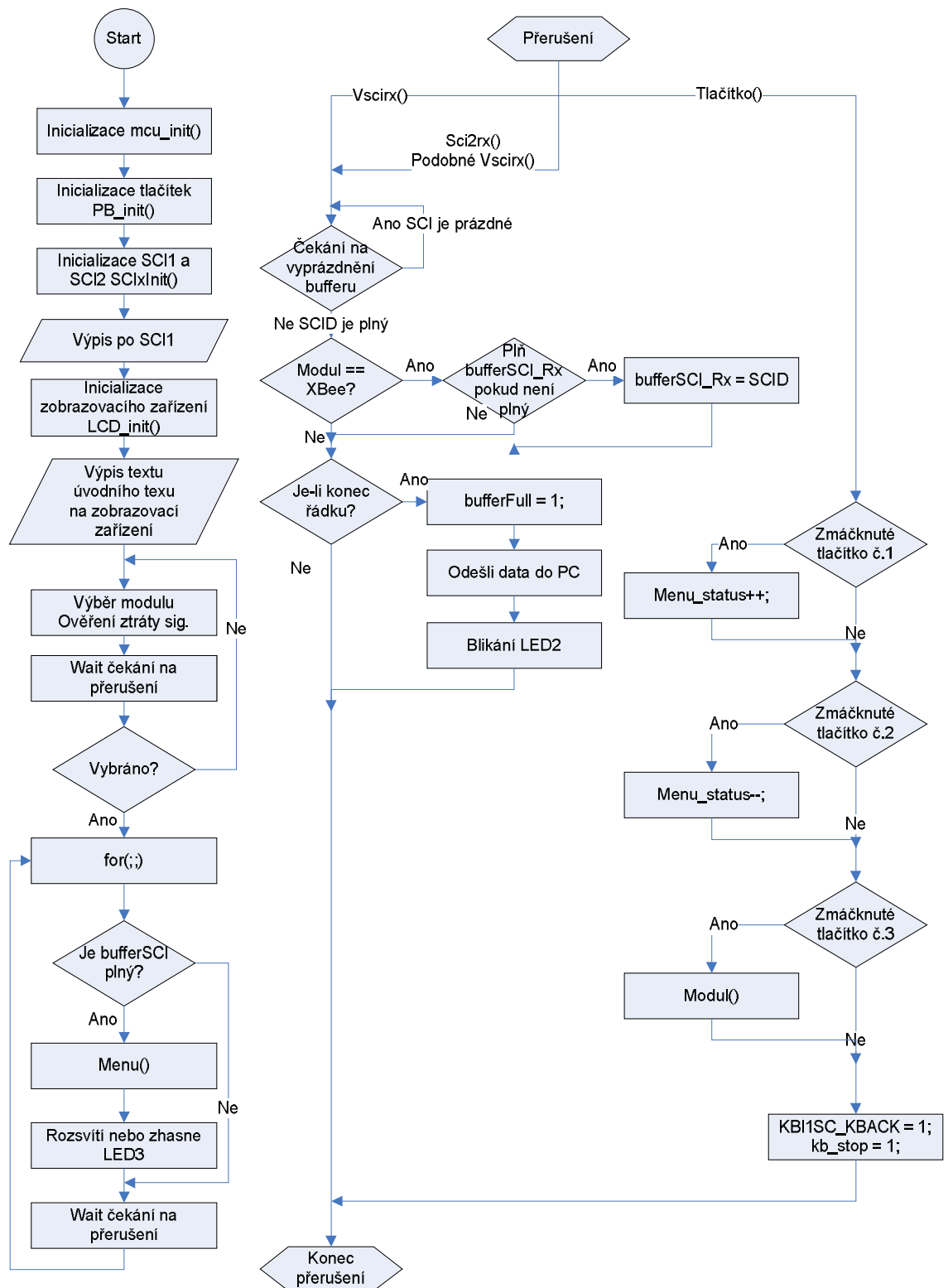
Bit name	Setting / Status	
I/D	0 = Decrement cursor position	1 = Increment cursor position
S	0 = No display shift	1 = Display shift
D	0 = Display off	1 = Display on
C	0 = Cursor off	1 = Cursor on
B	0 = Cursor blink off	1 = Cursor blink on
S/C	0 = Move cursor	1 = Shift display
R/L	0 = Shift left	1 = Shift right
DL	0 = 4-bit interface	1 = 8-bit interface
N	0 = 1/8 or 1/11 Duty (1 line)	1 = 1/16 Duty (2 lines)
F	0 = 5x7 dots	1 = 5x10 dots
BF	0 = Can accept instruction	1 = Internal operation in progress

Tabulka získaná s internetových stránek viz [3]

Příloha B

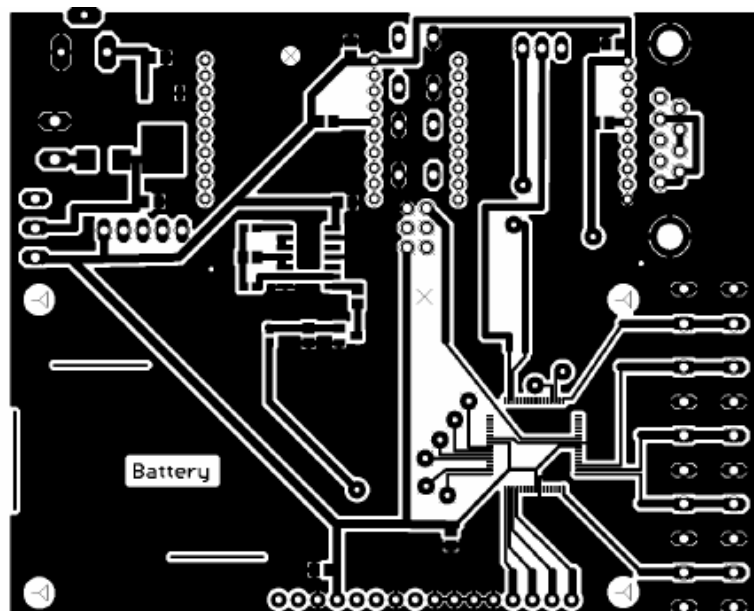
Tab. B: Vektory přerušení pro MC9S08GB60 viz [1]

Vector Priority	Vector Number	Address (High/Low)	Vector Name	Module	Source	Enable	Description
<div> <div>Lower</div> <div>↑</div> <div>↓</div> <div>Higher</div> </div>	26 through 31	\$FFC0/FFC1 through \$FFCA/FFCB	Unused Vector Space (available for user program)				
	25	\$FFCC/FFCD	Vrti	System control	RTIF	RTIE	Real-time interrupt
	24	\$FFCE/FFCF	ViiC1	IIC	IICIS	IICIE	IIC control
	23	\$FFD0/FFD1	Vatd1	ATD	COCO	AIEN	AD conversion complete
	22	\$FFD2/FFD3	Vkeyboard1	KBI	KBF	KBIE	Keyboard pins
	21	\$FFD4/FFD5	Vsci2tx	SCI2	TDRE TC	TIE TCIE	SCI2 transmit
	20	\$FFD6/FFD7	Vsci2rx	SCI2	IDLE RDRF	ILIE RIE	SCI2 receive
	19	\$FFD8/FFD9	Vsci2err	SCI2	OR NF FE PF	ORIE NFIE FEIE PFIE	SCI2 error
	18	\$FFDA/FFDB	Vsci1tx	SCI1	TDRE TC	TIE TCIE	SCI1 transmit
	17	\$FFDC/FFDD	Vsci1rx	SCI1	IDLE RDRF	ILIE RIE	SCI1 receive
	16	\$FFDE/FFDF	Vsci1err	SCI1	OR NF FE PF	ORIE NFIE FEIE PFIE	SCI1 error
	15	\$FFE0/FFE1	Vspi1	SPI	SPIF MODF SPTEF	SPIE SPIE SPTIE	SPI
	14	\$FFE2/FFE3	Vtpm2ovf	TPM2	TOF	TOIE	TPM2 overflow
	13	\$FFE4/FFE5	Vtpm2ch4	TPM2	CH4F	CH4IE	TPM2 channel 4
	12	\$FFE6/FFE7	Vtpm2ch3	TPM2	CH3F	CH3IE	TPM2 channel 3
	11	\$FFE8/FFE9	Vtpm2ch2	TPM2	CH2F	CH2IE	TPM2 channel 2
	10	\$FFEA/FFEB	Vtpm2ch1	TPM2	CH1F	CH1IE	TPM2 channel 1
	9	\$FFEC/FFED	Vtpm2ch0	TPM2	CH0F	CH0IE	TPM2 channel 0
	8	\$FFEE/FFEF	Vtpm1ovf	TPM1	TOF	TOIE	TPM1 overflow
	7	\$FFF0/FFF1	Vtpm1ch2	TPM1	CH2F	CH2IE	TPM1 channel 2
	6	\$FFF2/FFF3	Vtpm1ch1	TPM1	CH1F	CH1IE	TPM1 channel 1
	5	\$FFF4/FFF5	Vtpm1ch0	TPM1	CH0F	CH0IE	TPM1 channel 0
	4	\$FFF6/FFF7	Vicg	ICG	ICGIF (LOLS/LOCS)	LOLRE/LOCRE	ICG
	3	\$FFF8/FFF9	Vlvd	System control	LVDF	LVDIE	Low-voltage detect
	2	\$FFFA/FFFB	Virq	IRQ	IRQF	IRQIE	IRQ pin
	1	\$FFFC/FFFD	Vswi	Core	SWI Instruction	—	Software interrupt
	0	\$FFFE/FFFF	Vreset	System control	COP LVD RESET pin Illegal opcode	COPE LVDRE — —	Watchdog timer Low-voltage detect External pin Illegal opcode

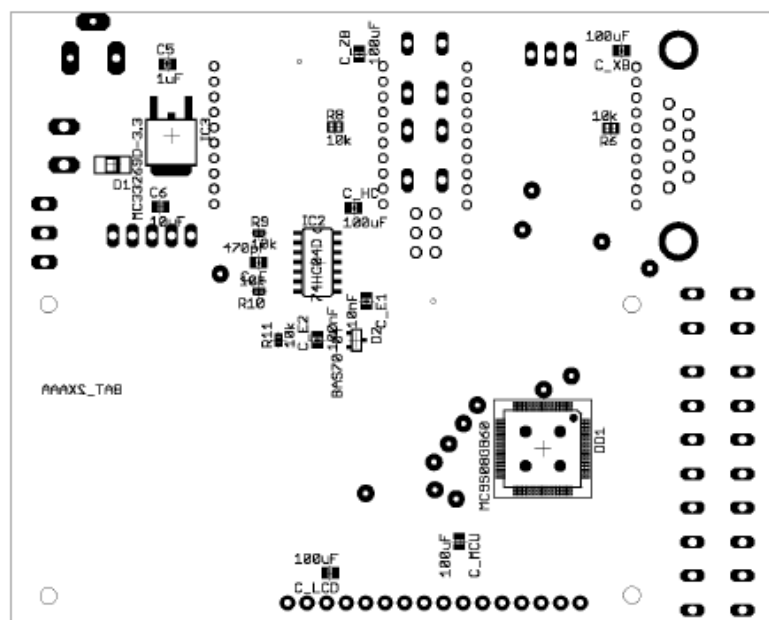


Obr. Příloha B: Vývojový diagram programu nahráného v řídicím procesoru MC9S08GB60

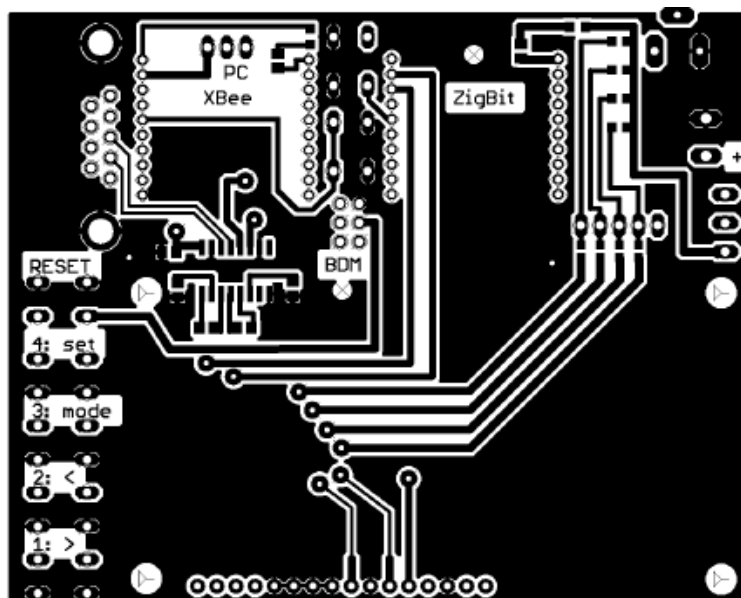
Příloha C



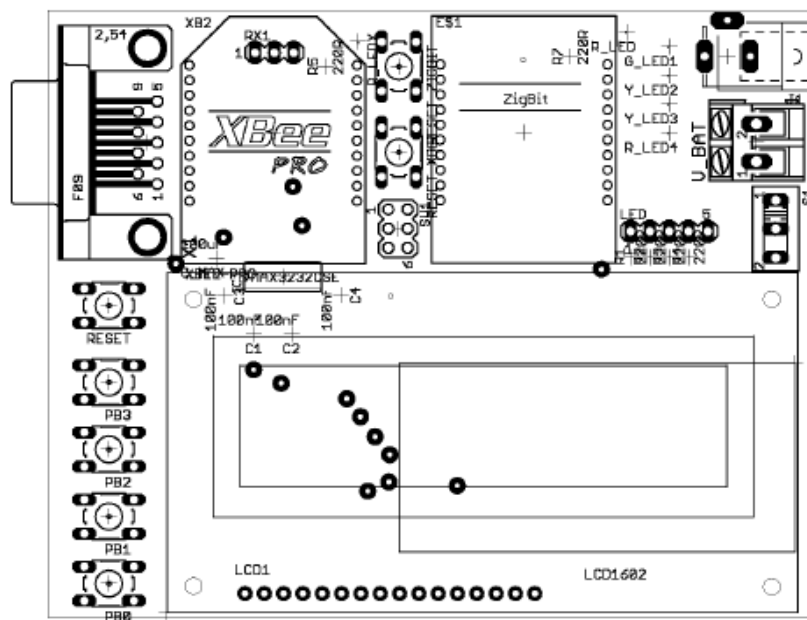
Obr. Příloha C: DPS - Bottom layer (rozměr 99 x 80 mm)



Obr. Příloha C: DPS - Bottom Layer - rozložení součástek

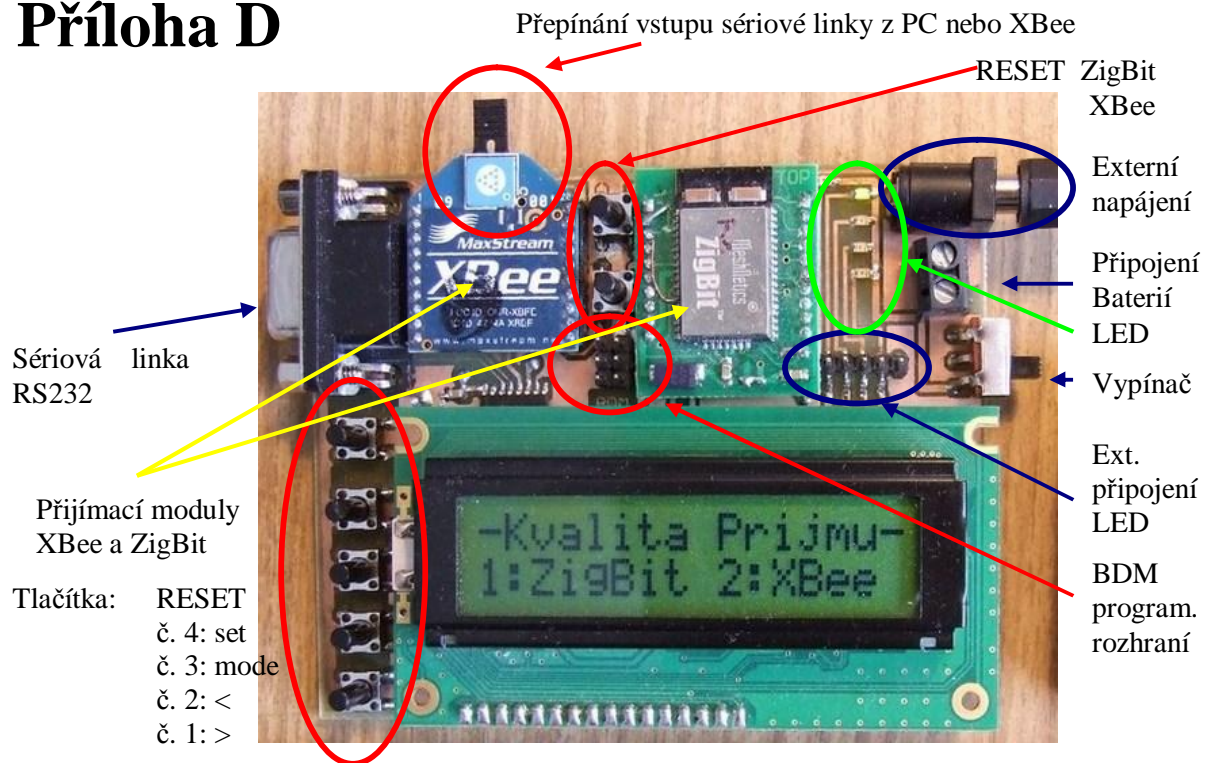


Obr. Příloha C: DPS - Top layer (rozměr 99 x 80 mm)

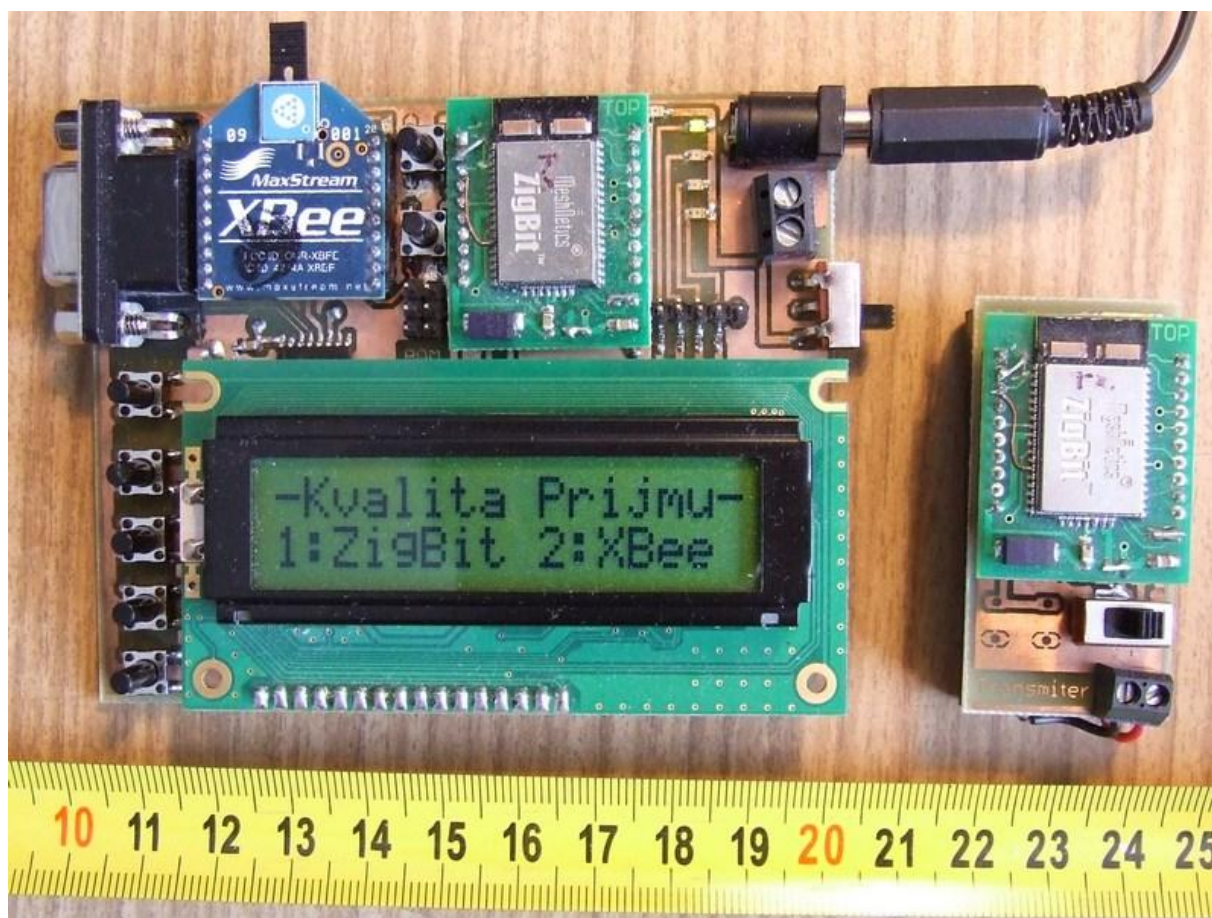


Obr. Příloha C: DPS - Top layer - rozložení součástek

Příloha D



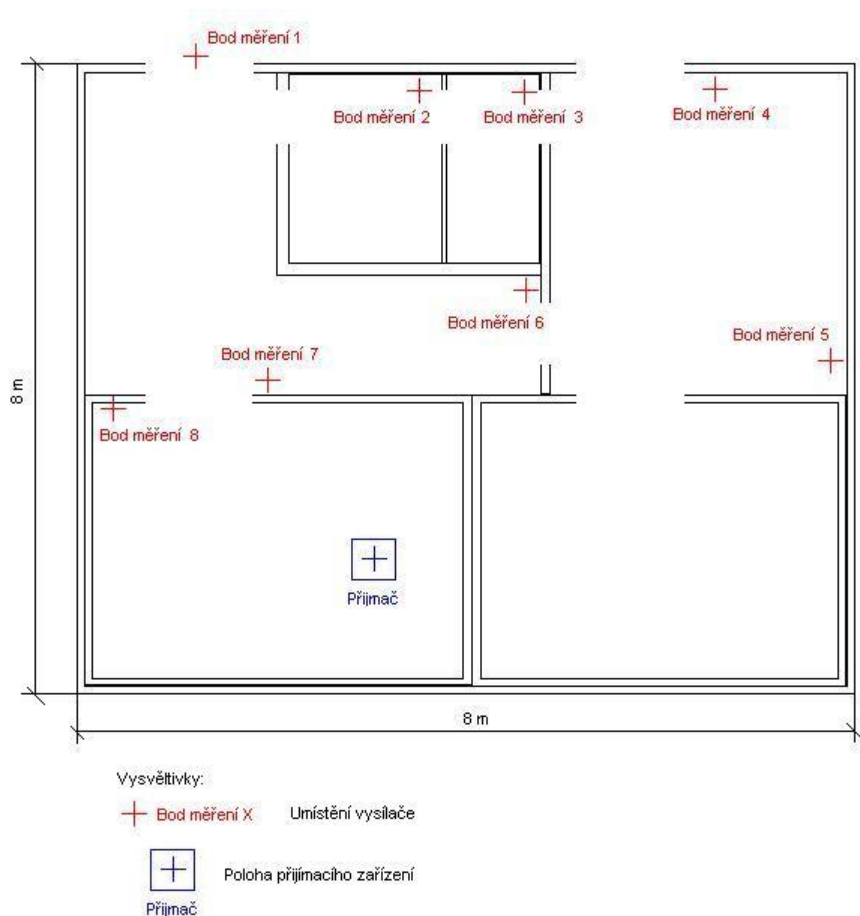
Obr. Příloha D: Popis přijímacího zařízení



Obr. Příloha D: Přijímací a vysílací zařízení

Příloha E

Příklad praktického měření v budově, při kterém si stanovíme počet přijatých paketu na cca 10000 a zaznamenáme naměřené hodnoty. Celkový počet měření bude 8, abychom zjistili vliv kvality signálu na umístění vysílače. Přijímací zařízení bude umístěno totožném místě.



Obr. Příloha E: Měřící místa v bytě a rozložení místností v bytě

Tab. C: Naměřené hodnoty

Bod měření	FC	FEC	BEC	BER	LQI	RSSI
	-	-	-	-	-	dBm
1	9980	0	0	0	255	-80
2	9967	0	0	0	255	-76
3	9971	0	0	0	255	-82
4	9954	7	560	$5,49 \cdot 10^{-5}$	254	-85
5	9992	0	0	0	255	-80
6	9977	0	0	0	255	-80
7	9993	0	0	0	255	-74
8	9973	0	0	0	255	-67

Během měření byly všechny dveře zavřené a v místnostech se nepohybovaly žádné osoby, kromě obsluhy u přijímacího zařízení. Všechna měření byla provedena pouze z dat získaných z přijímacího modulu ZigBit.

V dalším testu jsem měřil maximální možnou vzdálenost mezi přijímačem a vysílačem z hlediska možného stálého přenosu dat. Podmínka stanovená pro měření byla hledání maximální vzdálenosti, kde došlo ke ztrátě signálu a opětovném nalezení. Signál mohl několikrát během přijímání vypadnout. Další podmínka byla přijmutí 10000 paketů.

Tab. D: Test měření maximální vzdálenosti

Měření	Vzdálenost R <> T	FC	FEC	BEC	BER	LQI	RSSI
	[m]	-	-	-	-	-	[dBm]
1.	115	9967	5129	252025	0,024693	154	-89
2.	120	9980	4670	302310	0,029582	120	-88
3.	156	9971	4216	144253	0,014128	160	-89

Maximální vzdálenost v každém bodu měření je provedena z jiných míst měření a tím se signál mohl šířit různými cestami. Zajímavé je třetí měření, kde se podařilo měřit signál až vzdálenost 156 m. Měření probíhalo na velkém prostranství, kde byl větší počet zaparkovaných aut a pár vzdálených budov podél vzdálenosti přijímač vysílač. Ve všech bodech měření byla zajištěna přímá viditelnost mezi vysílačem a přijímačem. V testu měření maximální vzdálenosti jsem použil jen moduly ZigBit.